Advances in Applied Artificial Intelligence

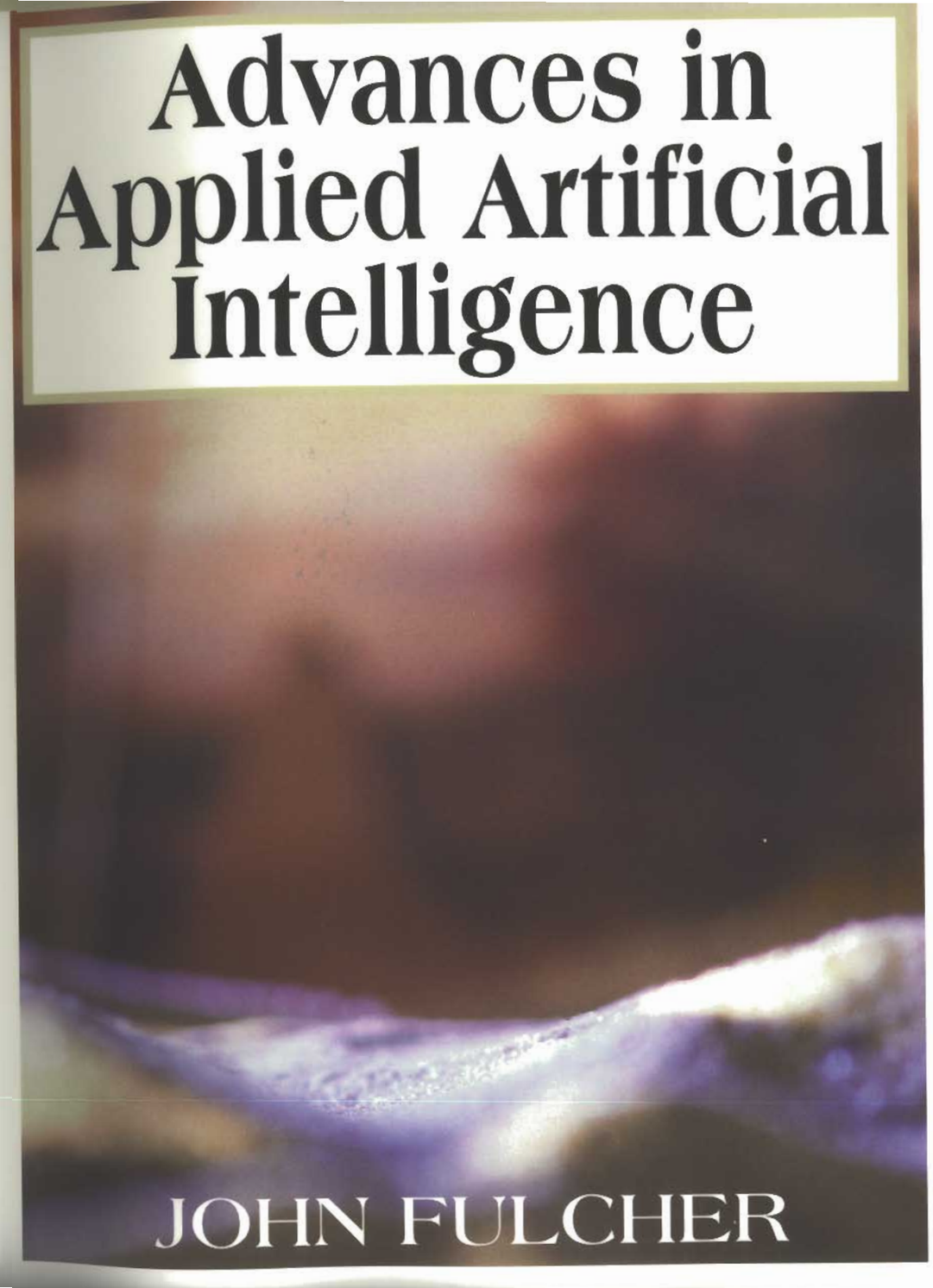
Whether any one technology will prove to be the central one in creating artificial intelligence, or whether a combination of technologies will be necessary to create an artificial intelligence is still an open question, so many scientists are experimenting with mixtures of such techniques. In Advances in Applied Artificial Intelligence these questions are implicitly addressed by scientists tackling specific problems which require intelligence in both individual and combinations of specific artificial intelligence techniques.

Advances in Applied Artificial Intelligence includes extensive references within each chapter which an interested reader may wish to pursue. Therefore, this book can be used as a central resource from which major avenues of research may be approached.

IDEA GROUP PUBLISHING
701 E. Chocolate Avenue - Suite 200
Hershey, PA 17033, USA
www.idea-group.com

$79.95
ISBN 159140828-8
57995
9 781591 408284

Fulcher — Advances in Applied Artificial Intelligence



Advances in Applied Artificial Intelligence

John Fulcher, University of Wollongong, Australia

IDEA GROUP PUBLISHING
Hershey • London • Melbourne • Singapore

This book is dedicated to
Taliver John Fulcher.

Advances in Applied Artificial Intelligence

Table of Contents

Chapter VII

Self-Organising Impact Sensing Networks in Robust Aerospace Vehicles

Mikhail Prokopenko, Geoff Poulton
Don Price, Peter Wang
Philip Valencia, Nigel Hoschke
Tony Farmer, Mark Hedley
Chris Lewis, and Andrew Scott
CSIRO Information and Communication Technology Centre and
CSIRO Industrial Physics, Australia

ABSTRACT

An approach to the structural health management (SHM) of future aerospace vehicles is presented. Such systems will need to operate robustly and intelligently in very adverse environments, and be capable of self-monitoring (and ultimately, self-repair). Networks of embedded sensors, active elements, and intelligence have been selected to form a prototypical "smart skin" for the aerospace structure, and a methodology based on multi-agent networks developed for the system to implement aspects of SHM by processes of self-organisation. Problems are broken down with the aid of a "response matrix" into one of three different scenarios: critical, sub-critical, and minor damage. From these scenarios, three components are selected, these being: (a) the formation of "impact boundaries" around damage sites, (b) self-assembling "impact networks", and (c) shape replication. A genetic algorithm exploiting phase transitions in systems dynamics has been developed to evolve localised algorithms for impact boundary formation, addressing component (a). An ant colony optimisation (ACO) algorithm, extended by way of an adaptive dead reckoning scheme (ADRS) and which incorporates a "pause" heuristic, has been developed to address (b). Both impact boundary formation and ACO-ADRS algorithms have been successfully implemented on a "concept demonstrator", while shape replication algorithms addressing component (c) have been successfully simulated.

INTRODUCTION

Structural health management (SHM) is expected to play a critical role in the development and exploitation of future aerospace systems, operating in harsh working environments and responding to various forms of damage and possible manufacturing and/or assembly process variations. SHM is a new approach to monitoring and maintaining the integrity and performance of structures as they age and/or sustain damage. It differs from the traditional approaches of periodic inspection and out-of-service maintenance by aiming for continuous monitoring, diagnosis, and prognosis of the structure while it is in service, damage remediation and, ultimately, self-repair. This requires the use of networked sensors and active elements embedded in the structure, and an intelligent system capable of processing and reducing the vast quantities of data that will be generated, to provide information about the present and future states of the structure, and to make remediation and repair decisions.

This chapter outlines an approach being taken to the development of next-generation SHM systems, and the development of a flexible hardware test-bed for evaluating and demonstrating the principles of the approach. This introductory section will outline the general requirements of an SHM system, provide an overview and relevant details of the hardware test-bed, and introduce our approach to the systems-level issues that must be solved.

Structural health management systems will eventually be implemented in a wide range of structures, such as transport vehicles and systems, buildings and infrastructure, and networks. Much of the current research effort is aimed at the high value, safety-critical area of aerospace vehicles. CSIRO is working with NASA (Abbott, Doyle, Dunlop, Farmer, Hedley, Herrmann et al., 2002; Abbott, Ables, Batten, Carpenter, Collings, Doyle et al., & Winter, 2003; Batten, Dunlop, Edwards, Farmer, Gaffney, Hedley et al., 2004; Hedley, Hoschke, Johnson, Lewis, Murdoch et al., & Farmer, 2004; Price, Scott, Edwards, Batten, Farmer, Hedley et al., 2003; Prokopenko, Wang, Price, Valencia, Foreman, Farmer, 2005a) and other key industry players to develop and test concepts and technologies for next-generation SHM systems in aerospace vehicles. While many of the principles of SHM systems described in this chapter are quite general, aerospace vehicles will be used throughout as example structures.

General Requirements of a Structural Health Management System

The key requirements of an advanced health monitoring system are that it should be able to detect damaging events, characterize the nature, extent, and seriousness of the damage, and respond intelligently in whatever timescale is required, either to mitigate the effects of the damage or to effect its repair. Strictly speaking, a pure monitoring system is expected only to report damage rather than to formulate a response, but it is preferable that the ultimate objective of responding to damage be borne in mind from the outset.

The statement of key requirements serves to sub-divide the problem as follows:

1. **Detection of damaging events**, which requires some knowledge of the environment in which the vehicle will be operating, the threats that it will face, and the development of sensors as well as a strategy for using them to detect damage events well within the time required for the system to respond. For events that require a rapid response, the best solution will often involve the use of passive, embedded sensors.
2. **Evaluation of the extent and severity of the damage**. This may or may not be a separate process from event detection. It may use different sensors, or the same sensors may be used in a different way. It is more likely to employ active sensors, which may be embedded in the structure or could be mobile and autonomous.
3. **Diagnosis of the damage**, which includes identification of the nature of the damage (for example, is it due to corrosion, fatigue, impact, and so on?) and its cause. An intelligent system should be able to utilize data from a vast array of sensors to deduce information about the events that have occurred and the resulting damage, on a whole-of-vehicle basis. Knowledge of the cause of damage may enable actions to be taken to reduce the rate of damage progression. Diagnosis also requires an assessment of the effect of the damage on the performance capability and integrity of the structure.
4. **Prognosis for the structure** requires prediction of the future progression of the damage and assessment of the effect of the forecast damage on structural performance. It requires an estimate of the future operating conditions of the structure.
5. **Formulation of the response: intelligent decision-making**. The nature of the response will depend on a number of factors such as the range of possible response mechanisms, the diagnosis of the damage (steps 2 and 3 above), the available response time as deduced from the diagnosis and prognosis (step 4), and so on. A response may consist of a sequence of actions. Major damage may demand an immediate emergency response, such as the rapid isolation of a whole section of the vehicle, followed by a more considered damage evaluation and repair strategy.
6. **Execution and monitoring of the response**. In addition to repair, a holistic response may involve changes to the flight or operational characteristics of the vehicle, either to mitigate the effects of the damage or to assist in the avoidance of further damage. The effectiveness of the response will require monitoring.

The first and second of these points are what is generally referred to as structural health monitoring. It is currently carried out in a very limited way in specific regions of selected structures (for instance, some aircraft, some items of large infrastructure), generally using a small number of sensors connected to a data logger or computer. Ultimately, large numbers of sensors will be required to detect and evaluate a wide range of possible damage types within a large and complex structure.

NASA's vision of self-monitoring robust aerospace vehicles includes both local and global SHM systems (Generazio, 1996). The local actions are anticipated to autonomously identify, evaluate, and trigger an appropriate response, including repair, for a wide range of damage or defect conditions in aerospace materials and structures, using distributed micro-sized sensors, multiple miniature robotic agents, micro-sized repair tools, and self-healing smart structures. In parallel, global actions should enable dynamic evaluation of structural integrity across large and remote areas. This dual architecture, in turn, entails the need for dynamic and decentralised algorithms used in all the key requirements enumerated above.

An additional key requirement of an autonomous SHM system is robustness. The system must be able to operate effectively in the presence of damage to the structure and/or failure of system components: its performance must degrade "gracefully" rather than catastrophically when damage occurs. Scalability, reliability, and performance verification are also needed.

Also of great importance to any SHM system is the provision of an efficient and robust communications system. Unless local actions are sufficient, the key requirements mentioned above will rely on communication from a damage site to another part of the vehicle, for example, to initiate secondary inspections, repair, or in extreme cases, appropriate emergency action. Such communications will most likely be hierarchical and flexible, since the site to which damage is reported will vary with time, as well as the damage location and severity. Robustness must also be a feature, with continuing communications ensured even in severe damage situations.

In order to address these requirements, we have chosen to apply a multi-agent system (MAS) approach to the architecture, and seek to develop design methodologies that will enable the desired responses of the system (the remedial actions) to emerge as self-organised behaviours of the communicating system of sensing and acting agents. The particular MAS structure which is the focus of this chapter is a group of contiguous agents, locally connected and forming the surface of a three-dimensional object. Each agent has sensing and computational capabilities, and can communicate only with its immediate neighbours. Thus all communications, local, regional, and global need to occur through these agent-to-agent links. Although such constraints impede the flow of information, there is a significant potential redundancy which can aid robustness. Various types of communications will be needed, ranging from local cell-to-cell handshaking to check status, to emergency global communications in case of severe damage, which must be carried out as rapidly as possible whenever needed.

Much of this research has been undertaken as part of the CSIRO-NASA Ageless Aerospace Vehicle (AAV) project, which also includes an experimental test-bed and concept demonstrator (CD) system, whose aim is to detect, locate, and evaluate impacts by fast particles. A software simulation package has also been developed. The purpose of these two tools is to provide versatile research platforms for investigations of sensing, data processing, communications, and intelligence issues, and for demonstrating solutions for some of these issues. The architecture of the system is highly modular, being composed of "cells" that constitute the outer skin of the vehicle. Each cell consists of

a small region of the vehicle skin, a number of sensors attached to the skin, a processing unit, and communication ports. Each of these cells is an agent in the multi-agent system architecture. This system will be described in more detail in the *Ageless Aerospace Vehicle Project* section.

The Approach to Intelligent SHM System Development

The approach adopted here to the development of system intelligence is based on a multi-agent system (Ferber, 1999) in which the desired responses emerge by self-organisation. What is meant by self-organisation? The following definition, in the context of pattern formation in biological systems, was given by Camazine, Deneubourg, Franks, Sneyd, Theraulaz, and Bonabeau, (2001):

> *Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.*

This definition captures two important aspects of self-organisation. Firstly, the global behaviour of the system of many interacting components (agents) is a result only of the interactions between the agents, and secondly, that the agents have only local information, and do not have knowledge of the global state of the system. Typically, this emergent behaviour at the system level is not easily predictable from local agents' rules and interactions.

Self-organisation occurs in both biological and non-biological systems. In non-biological systems, self-organisation is produced by a flow of energy into the system that pushes it beyond equilibrium: the winds that produce characteristic ripples in sand, the temperature gradients that produce Bénard convection cells in a viscous fluid, the thermodynamic forces that lead to crystal growth and characteristic molecular conformations are all examples of these external energy inputs. However, the nature of the emergent behaviour depends critically on the interactions between the low-level components of the systems — the grains of sand, the molecules in the fluid, the atoms in the crystals and molecules. These interactions are determined by the laws of nature and are immutable.

In biological systems, on the other hand, the interactions between components of a system may change over generations as a result of evolution. There are selection pressures based on adaptation to the environment and survival. These selection pressures lead to emergent behaviour that is desirable for the survival of the system in the environment in which it has evolved, but which may be undesirable in other environments. Similarly, when using evolutionary methods for the design of complex SHM systems that employ self-organised responses to damage, there is a need to identify appropriate selection pressures. These, through their contribution to an evolutionary fitness function, will constrain the agent interactions to produce desirable emergent responses. Such selection pressures will be further discussed later in this chapter.

Current approaches developed for complex systems, and in particular, multi-agent networks, either solve individual problems using evolutionary algorithms, or restrict the

solution space so that emergent behaviour is impossible. Both these approaches are inadequate, the first because of high computational needs and the loss of an intuitive feel for the results, and the second because it is likely to over-constrain the range of possible solutions: it is noted that biological systems make extensive use of the rich solution space provided by the complexity of natural systems. In order to emulate this capability, a general design methodology, retaining the essential complex behaviour of multi-agent systems, is needed. Design in this context means the ability to specify the local agent properties so that they interact to produce a desired global result.

In this chapter, we describe an initial hybrid top-down/bottom-up (TDBU) attempt at subdividing a set of high-level goals into intermediate hierarchical objectives, and exploring the solution space at each intermediate level of the hierarchy. In particular, we explicitly define the main functional SHM sub-tasks, working downwards from the top-level design goals. The next stage is, for each sub-task, to design localised algorithms working from the bottom up and using an iterative process including the following steps:

1. forward simulation leading to emergent behaviour for a task-specific class of localised algorithms;
2. quantitative measurement of desirable qualities shown by the emergent patterns (for example, spatiotemporal stability, connectivity, and so on); and
3. evolutionary modelling of the algorithms, with the metrics obtained at step (b) contributing to the fitness functions.

While the eventual optimal solution to the overall SHM problem may not involve sequential steps through the sub-tasks listed above, our initial approach is to divide the problem along the lines indicated. Thus we will first aim to develop procedures to characterise damage (in terms of its nature, location, extent, and severity), then form a diagnosis, then a prognosis, and finally make decisions and take appropriate actions.

A diagnosis, or the confidence in a diagnosis, may change with time, as the development of damage is monitored and more information becomes available. One of the major benefits of SHM is the ability it provides to detect damage at an early stage and to monitor its development, leading to improved diagnostic capabilities and, ultimately, more efficient repair strategies. Similarly, a prognosis, which depends on prediction of the future progression of the damage, can be modified with time as the damage develops.

The Response Matrix Approach to Comparing Response Characteristics

It is clear that an intelligent system in a safety-critical environment must be able to respond very differently in different circumstances. In the event of sudden critical damage, such as a major impact, the most important characteristic of the response may be speed. Some undesirable side effects may be a tolerable trade-off for a rapid and effective emergency response. On the other hand, an acceptable response to slowly developing non-critical damage, such as highcycle fatigue or corrosion, must be more deliberative and targeted, and response speed is unlikely to be a relevant consideration. In order to provide a basis for comparison of response types, and to guide thinking about the processes by which responses are produced, the following simple response matrix method has been developed.

The response matrix seeks to classify a response on the basis of its spatial extent and the degree of deliberation required to form the response. The spatial extent is defined in terms of system cells, where a cell is the smallest intelligent unit of the system. Examples of cells in the AAV test-bed are described in the *Ageless Aerospace Vehicle* section. A response is categorized as "local" if only a single cell is involved, as "neighbourhood" if only a small group of neighbouring cells is involved, or as "global" if a larger region, such as a complete sub-structure or even the whole structure is involved in the response. The nature of a response is considered to be "reactive" if it is made rapidly, using only the initially-sensed data, and with effectively no feedback that could be classed as deliberation. It is said to be "strongly deliberative" if there are long feedback loops involved in obtaining additional sensed data, and making a response that would be classified as being highly intelligent. A "moderately deliberative" response would involve some deliberation by the system, but with shorter feedback loops than required for a strongly deliberative response. Some examples of these responses will be outlined below to clarify these definitions; but first, three levels of damage will be defined.

Three levels of damage will be referred to throughout this chapter. The first is *critical damage*, which is sufficiently severe to threaten the integrity of the structure, and possibly the survival of the vehicle. Critical damage will generally occur suddenly, or presumably its precursors would have been detected and corrected. It will require an emergency response that must be rapid and effective, even if subsequent, more thorough diagnosis shows it to have been an over-reaction. The second level is *sub-critical damage*, which, although severe enough to require an immediate response, is not sufficiently threatening to the vehicle's survival to require an emergency response. Thirdly, there is *non-critical, or minor damage*, which does not necessarily require an immediate response, but which must be monitored to track its progression with time (as with, for example, corrosion or fatigue damage), or its possible interaction with other damage mechanisms.

In terms of these levels of damage, a reactive response will generally be invoked only by critical damage, or by an indication of the likelihood of critical damage: it will generally be preferable to react to the likelihood of critical damage than not. A reactive response is pre-programmed (such as an emergency evacuation from a building), and will be followed by a more deliberative evaluation and diagnosis. It may include physical and/or electronic isolation of a cell, neighbourhood, or sub-structure, and the initiation of autonomic and fast temporary repairs. The response to indications of sub-critical or non-critical damage will be to evaluate the severity of the damage, by monitoring the outputs of sensors other than those that indicated the damage, or by initiating active damage evaluation (ADE) using either embedded or mobile sensors. The ADE and subsequent remedial actions may be moderately deliberative or strongly deliberative, depending on the amount of information and prior knowledge which is required for a diagnosis. A moderately deliberative response might consist of a rapid diagnosis from a single set of ADE data, followed by an immediate remedial response. On the other hand, an accurate diagnosis of non-critical damage might require the damage progression to be monitored for some time, or it may require several sets of ADE data and comparison with a physical damage model, and this would be considered a strongly deliberative response.

Examples of the ways in which chains of responses to different types and levels of damage can be classified using the response matrix approach are shown in Table 1. Two

forms of potentially critical damage, a fast particle (perhaps a micro-meteoroid) impact on a single cell, and large body impact or an explosion that causes severe damage to a whole sub-structure, as well as one of non-critical damage to one or more cells, are considered. In the cases of critical damage, the initial reactive response that is required

Table 1. The response matrix approach for classification of system responses, with examples for three types of damage, as outlined in the text

Degree of Deliberation	Local	Neighbourhood	Global
Reactive	Major damage to single cell by fast particle impact. – Ignore messages from damaged cells. – Report damage to neighbourhood action point(s). – Initiate automatic temporary seal. Follow by Moderately Deliberative, Neighbourhood response.		Major damage to sub-structure by large body impact or by explosion or fire. – Initiate emergency comms. to global action site. – Physically and electronically isolate sub-structure. Follow by Moderately Deliberatively, Global response
Moderately Deliberative		– ADE by neighbouring cells to assess damage to cell. – If damage not critical, and/or if local diagnostics are favorable, re-enable cell. – If damage critical, initiate cell replacement.	– ADE by remote sensors, mobile sensors or robotic swarm. – Identify damaged region. – Re-enable cells that are not critically damaged. Follow by Strongly Deliberative, Global response
Strongly Deliberative	Corrosion or degradation damage to single cell. – Monitor damage and environmental effectors as damage progresses.	– Set up damage network in neighbourhood. – Compare with damage models or prior knowledge to develop diagnosis and prognosis. – Take remedial action when indicated by prognosis.	– Diagnosis and prognosis of damaged region. – Respond as indicated by prognosis: repair or replace damaged cells if necessary, monitor state of others.

Spatial Extent

to ensure survival is followed by a more deliberative response to obtain more specific information about the damage, to produce more appropriate long-term remediation of the damage, and to enable the system to learn to deal better with similar events in the future. Similar sequences of reactive and deliberative response to danger ("panic" response) and damage can be recognized in animals, including humans.

Top-Down/Bottom-Up Design (TDBU) and the Response Matrix

One way of viewing the response matrix is as the top-down part of the TDBU approach to design which was outlined in the *Approach to Intelligent SHM System Development* section. For a range of damage scenarios and the desired system response to each, the response matrix infers the large-scale components necessary for the appropriate response to occur. The components, whilst not unique, are chosen as high-level and as broad in spatial extent as possible, in line with the minimal hierarchical decomposition of the problem which is the intention of the TDBU approach.

It is envisaged that most if not all of the components would be implemented by self-organisation within the multi-agent structure. If the decomposition is too broad, then there may be difficulty in achieving such self-organising solutions, while if it is too prescriptive, then the result may be an unsatisfactory system outcome. Of course, the ideal would be to achieve complete, self-organised responses to all likely damage scenarios without having to decompose the problem. The possibility of this is unlikely, at least in the near future, because of the complexity of multi-agent systems, so a minimal hierarchical decomposition is a good compromise.

The most important components are damage detection, local assessment of damage, higher-level assessment (diagnosis and prognosis), and response (actions). Some of these will be treated in detail in later sections. However, there is one critical component which cannot be left out of the equation. This is communications, which is a necessary part of all damage scenarios, and at least as complex and difficult to handle properly as the others mentioned above.

Communications

Agents in a multi-agent system communicate either directly or through the environment (stigmergy) to form a network which usually exhibits complex behaviour (Holland & Melhuish, 1999). If the agents are fixed in space, as on the skin of an aerospace vehicle, then direct inter-agent communications forms the basis of the network. As will be seen, the particular agent networks of interest here generally only support communications between adjacent neighbours. Although this appears at first glance to be a restriction, it is also the main source of network robustness because of the large redundancy provided by the network.

There are many different communications tasks which the network has to be able to handle, ranging from simple status queries and responses for adjacent neighbours to the reporting of a critical damage situation to a remote site from which appropriate action can be initiated. Although communications tasks vary with the damage scenario, they all share the need to transfer information robustly and efficiently from one part of the network to another, in an environment where both transmit and receive sites may not know the other's location (which may change with time anyway). In addition, the

environment itself is time-variable, particularly in times of significant damage. Secure communications in such an environment is a general task of significant difficulty which, however, needs to be solved since the survival of the vehicle may depend on it.

In a multi-agent system it would seem natural to seek self-organising solutions to this problem, and some progress has been made in this area, including one application presented in the *Impact Boundaries* section.

BACKGROUND AND RELATED WORK

Self-organisation is typically defined as the evolution of a non-equilibrium system into an organised form in the absence of external pressures. Over the last years, a number of examples employing self-organisation have been suggested in the broad context of biological and bio-inspired multi-agent systems: the formation of diverse spatial structures by groups of ants (Deneubourg & Goss, 1989), the growth and morphogenesis of networks of galleries in the ant *Messor sancta* (Buhl, Deneubourg, & Theraulaz, 2002); a propulsive motion of locally-connected mobile automata networks, dynamically organising into simple spatial structures while evolving toward task-specific topologies (Wessnitzer, Adamatzky, & Melhuish, 2001); a pattern formation of self-assembling modular robotic units, with the emergent chaining behaviour being analogous to the process of polymerisation, and the emergent clustering behaviour being similar to the autocatalytic process used by pheromone-depositing bark beetle larvae (Trianni, Labella, Gross, Sahin, Dorigo, & Deneubourg, 2002); fault-tolerant circuit synthesis on a self-configurable hardware platform provided by the Cell Matrix approach (Durbeck & Macias, 2002); a self-assembly of network-like structures connecting a set of nodes without using pre-existing positional information or long-range attraction of the nodes, using Brownian agents producing different local (chemical) information, responding to it in a non-linear manner (Schweitzer & Tilch, 2002).

Traditional multi-component systems do not exhibit self-organisation; instead, they rely on fixed multiple links among the components in order to efficiently control the system, having fairly predictable and often pre-optimised properties, at the expense of being less scalable and less robust. In the SHM context, condition-based maintenance (CBM), a process where the condition of equipment is automatically monitored for early signs of impending failure, followed by diagnostics and prognostics, has become popular for multi-component systems due to its cost and reliability advantages over traditional *scheduled* maintenance programs. However, according to a NASA Jet Propulsion Laboratory (JPL) report on Prognostics Methodology for Complex Systems (Gulati & Mackey, 2003), CBM is frequently difficult to apply to complex systems exhibiting emergent behaviour and facing highly stochastic environmental effects. A scalable solution capable of providing a substantial look-ahead capability is required. The JPL solution involves an automatic method to schedule maintenance and repair, based on a computational structure called the informed maintenance grid, and targeting the two fundamental problems in autonomic logistics: (1) unambiguous detection of deterioration or impending loss of function, and (2) determination of the time remaining to perform maintenance or other corrective action based upon information from the system (Gulati & Mackey, 2003). The solution based on the JPL work does not account for self-organisation and is not directly applicable to distributed multi-agent networks.

A recent paper by Prosser, Allison, Woodard, Wincheski, Cooper, Price, Hedley, Prokopenko, Scott, Tessler, and Spangler, (2004) has given an overview of NASA research and development related to SHM systems, and has discussed the requirements for SHM systems architectures. Characteristics such as scalability, flexibility and robustness were identified as being important requirements. Biological systems, including those referred to above, provide many examples of these characteristics in self-organising multi-agent systems. Indeed, it has been asserted that biological complexity and self-organisation have evolved to provide these characteristics. For example, Klyubin, Polani and Nehaniv (2004) indicated that evolution of the perception-action loop in nature aims at improving the acquisition of information from the environment and is intimately related to selection pressures towards adaptability and robustness — their work demonstrated that maximisation of information transfer can give rise to intricate behaviour, induce a necessary structure in the system, and ultimately be responsible for adaptively reshaping the system. In order to investigate the practical implementation of biologically-inspired concepts to structural health management systems, an experimental multi-agent test-bed has been developed. This will be described in the next section.

THE AGELESS AEROSPACE VEHICLE PROJECT

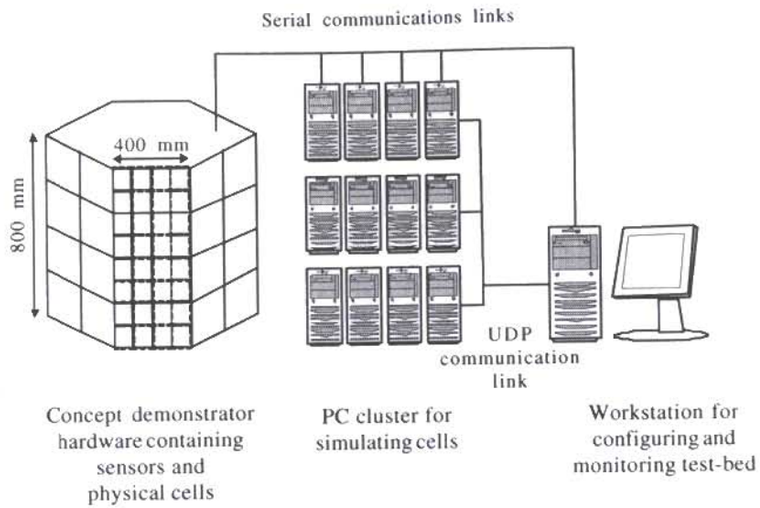
Introduction

The CSIRO-NASA Ageless Aerospace Vehicle (AAV) project has developed and examined concepts for *self-organising* sensing and communication networks (Abbott et al., 2002; Abbott et al., 2003; Price et al., 2003; Batten et al., 2004; Hedley et al., 2004; Prokopenko et al., 2005a). These concepts are being developed, implemented, and tested in an experimental test-bed and concept demonstrator: a hardware multi-cellular sensing and communication network whose aim is to detect and react to impacts by high-velocity projectiles that, for a vehicle in space, might be micro-meteoroids or space debris. High-velocity impacts are simulated in the laboratory using short laser pulses and/or steel spheres fired using a light-gas gun.

The test-bed has been built as a tool for research into sensor design, sensing strategies, communication protocols, and distributed processing using self-organising multi-agent systems. It has been designed to be modular and highly flexible: By replacing the sensors and their associated interface and data acquisition electronics, the system can be readily reconfigured for other applications.

Figures 1 and 2 contain a schematic overview of the system and photographs of its physical implementation, respectively. The physical structure is a hexagonal prism formed from a modular aluminium frame covered by 220 mm x 200 mm, 1-mm thick aluminium panels that form the outer skin of the structure. Each such panel contains four "cells", and each of the six sides of the prism contains eight of these panels. The skin therefore consists of 48 aluminium panels and 192 cells. Cells are the fundamental building blocks of the system: they are the electronic modules containing the sensing, processing, and communication electronics. Each cell occupies an area of ~100 mm x 100 mm of the skin, mounted on the inside of which are four piezo-electric polymer (PVDF)

Figure 1. Architecture of the test-bed



sensors in a 60 mm square array, to detect the acoustic waves that propagate through the skin as a result of an impact.

Each cell contains two electronic modules (Figure 2), one of which acquires data from the sensors, while the other runs the agent software and controls the communications with its neighbouring cells. Importantly, a cell communicates only with four immediate neighbours. The test-bed does not employ centralised controllers or communication routers.

Also shown in Figure 1 are a PC cluster and a workstation. The cluster is used to simulate a larger network of cells, and is used for research into the emergent behaviour of multi-agent algorithms in very large networks. The workstation is used to initialise and configure the test-bed, and to monitor the network during operation, for visualization and debugging purposes. However, it is not part of the sensor network and does not influence or control the system behaviour during normal operation. This workstation is the "System Visualization" block shown in Figure 3 (upper right), which is a schematic diagram of the multi-agent system architecture of the test-bed system.

A picture of the current state of the physical test-bed, with some panels removed to reveal the internal structure and electronics, is shown in Figure 2. A 12V power supply is mounted on the base of the hexagon, and power is distributed via the top and bottom edges. Communication between the test-bed and PCs is via 1.5 Mbits/s serial links using USB.

Figure 2. Photograph of the test-bed (top); a single cell consisting of a network application sub-module (NAS) and a data acquisition sub-module (DAS) (center photograph); the bottom photograph shows an aluminium panel containing four cells

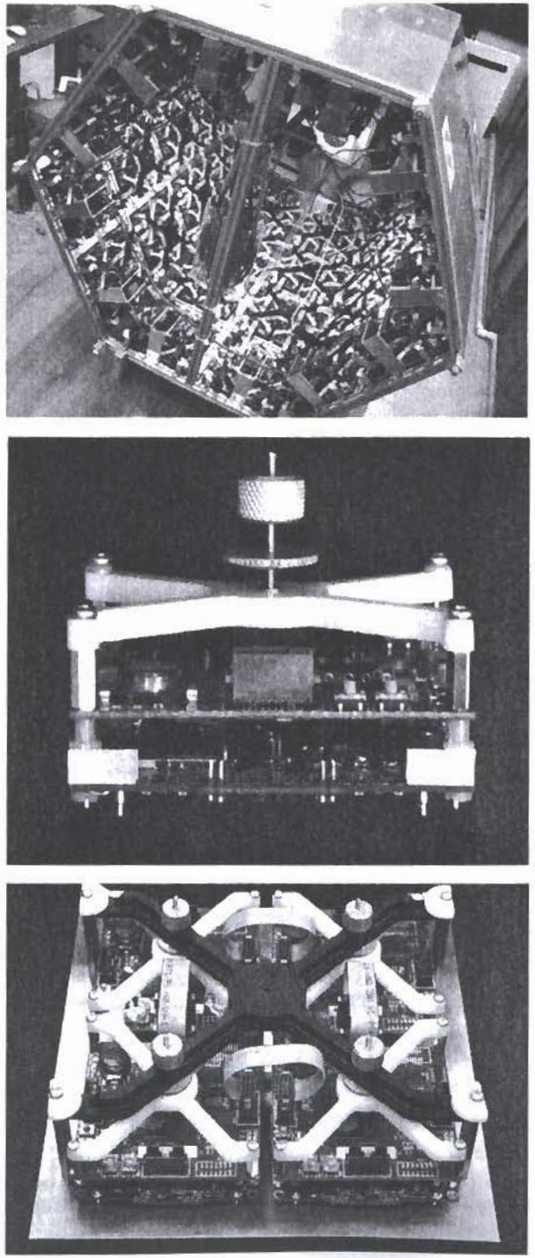
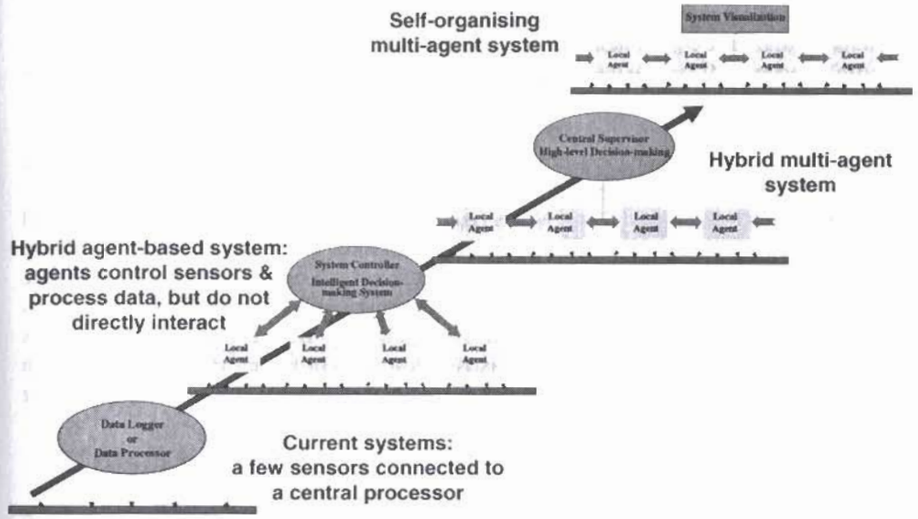


Figure 3. Schematic diagram of a progression of agent-based system architectures, leading to the complex multi-agent architecture of the test-bed system in the upper right of the diagram. Each cell in the test-bed is a local agent, capable of obtaining local information about damage from its sensors, and capable of communicating only with its direct neighbours in the mesh network. The workstation referred to in the text is the System Visualization block.



Cell Properties and Single-Cell Functionality

A modular approach to the sensing and electronics was adopted to enhance flexibility, re-configurability and ease of manufacture (Hedley et al., 2004, Batten et al., 2004). Each module, or cell, contains sensing elements, signal processing (analogue and digital), and communications, using the following logical layering of these functions:

1. **Sensors** — piezo-electric polymer sensors attached to the aluminium skin.
2. **Analogue signal processing** — including amplification, filtering, and other processing of the sensor signals required prior to digitization of the signals.
3. **Sampling and data pre-processing** — including digitization, calibration correction, data reduction, and other processing that can be performed using only the local signals.
4. **Data analysis and localised agent algorithms** — at this level, data is processed using information from local sensors and neighbouring modules.
5. **Inter-module communication** — comprising the software stack and the physical links to provide communication between modules.

These layers are divided into two groups, the data acquisition layer (DAL), which consists of layers 1 to 3, and the network application layer (NAL), which consists of layers 4 and 5. These are implemented as separate physical sub-modules, called the Data Acquisition Sub-module (DAS) and network application sub-module (NAS), respectively. This separation allows replacement of one type of sensor, and its associated electronics, with another sensor sub-module, without changing the main processing and communications hardware, hence allowing a range of sensor types to be tested.

The DAS provides gain and filtering for the four attached piezo-electric sensor signals (which have components up to 1.55 MHz after analogue filtering). These signals are digitized at 3 Msamples/sec using a 12-bit analogue-to-digital converter, and initial processing to estimate the time of arrival of a signal on each sensor is performed using digital signal processing. This information is passed to the NAS using a high-speed synchronous serial communications link, and power is received from the NAS over the same connector.

The NAS contains both a 400 MIPS fixed-point digital signal processor (DSP) and 400k gate field programmable gate array (FPGA), along with 2 Mbytes of non-volatile memory and 8 Mbytes of volatile memory. These resources are used by the software agents running in each cell, which provide the network intelligence. Each NAS contains a unique 64-bit identifier. An NAS contains four ports, used for communication with its four nearest neighbours and for power distribution. This provides a highly robust mesh network structure that will maintain connectivity even if a significant number of cells or communications links are damaged.

Impact Detection

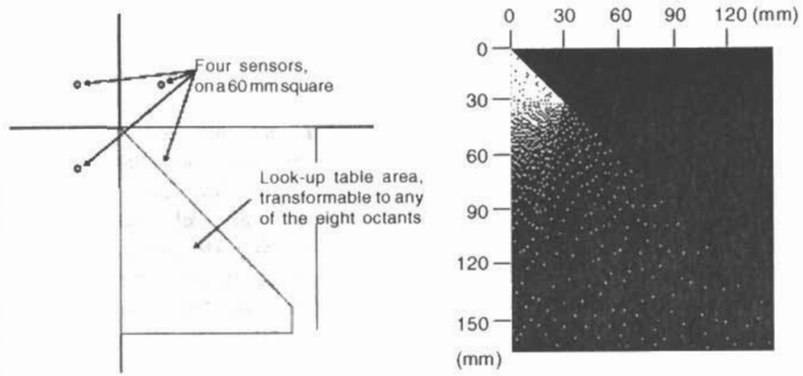
Piezo-electric sensors, consisting of a 2.5 mm-diameter, 110 µm-thick film of PVDF (polyvinylidene fluoride) coated on both sides with a conductive gold layer, are bonded to aluminium sheets which form the "skin" of the concept demonstrator, providing a suitable method for detecting the plate waves in the aluminium sheets.

The DAS continuously samples four analogue channels, storing the data in a circular buffer containing 200 samples, or 64 µs of data, from each channel, and checking if a sample has deviated by more than 90mV from the channel's average value. Once a channel has exceeded this selected threshold, a further 184 samples are taken on each channel, an impact is flagged, and the buffers are processed.

The signals from the four sensors are detected by the DAL electronics, narrow-band filtered, amplified, and digitized. Then the *earliest arrival time* is subtracted from the other three times to give three time delays. These delays are used to estimate the location of the impact relative to the centre of the square formed by the four sensors used. In other words, the impact is triangulated based on measured arrival times of the lowest-order extensional wave and the known group velocity (about 5300 m/s) of these waves in the aluminium plate at a particular frequency (about 1.5 MHz).

If standard triangulation techniques were applied to these three time delays, the equations for three hyperbolae would need to be solved simultaneously. Further, as the digitization rate is limited to around 3 MHz in the present hardware, the time of arrival of the extensional waves may not be determined accurately enough for solutions to these equations to exist. Searching for near solutions can be done, but this would take a significant amount of processor time and memory in the present configuration.

Figure 4. Left — Position of sensors and the area of the panel covered by the look-up table (the table is transformed to the other eight octants to cover the full panel); Right — Graphical representation of look-up table points and estimates of the positional errors (the top left-hand corner represents the centre of the group of four sensors, and the axes' labels are millimetres from the centre of the square formed by the sensors. The white dots are the points in the look-up table: Their spacing is an indication of the uncertainty in the estimate of the location of an impact).



Instead, the present version of the demonstrator employed a faster way: *a look-up table*. As each cell has four sensors arranged at the corners of a 60 mm square, the look-up table will be the same for each of the eight octants whose origin is at the centre of the square. For the present size of aluminium panel, the largest area that needs to be covered by the look-up table is a truncated, isosceles (at 45 degrees) right triangle (Figure 4), 165 mm long and 143 mm in the truncated direction. This geometry will cover the "worst case" of an impact in a corner of the panel diagonally opposite that of the cell containing the four sensors being used. This area was then divided into 1-mm squares. For each square, the time delays may be calculated and used to form a six-figure "index" that is associated with each position. This index number is formed from the three delay times from each pair of sensors, the first two digits being the shortest delay time (in number of time step intervals of 320 ns), the second two-digit number is the next longest delay time, and the final two-digit number the longest delay time. While the total number of points (on a 1-mm grid) inside this truncated triangular area is about 13,600, the total number of unique points (points that have different index numbers) is only 1,071 due to the finite time-step interval. The mean position of all the 1-mm cells that have the same index number is stored in the look-up table with that index number. The look-up table is stored in the flash memory on the DSP.

When an impact is registered, the delays are first ordered to quickly determine which octant of the Cartesian plane contained the impact site. Depending on the order of the impacts (the time of arrival at each of the sensors) one of eight co-ordinate transformation

matrices is used to translate the look-up table result to the correct orientation. The index number of the impact is then found in the look-up table, and the average position associated with that index number multiplied by the appropriate transformation matrix is given as the location of the impact. In this way, any of the four sets of four sensors on a panel can detect the location of an impact.

DAMAGE SCENARIOS

Introduction

In the *Introduction* section we outlined the processes associated with a damage situation and categorised them in terms of a matrix whose two variables are spatial extent and degree of deliberation. This is a convenient means of describing a wide range of damage situations by breaking them down into components which range in extent from local to global and in the associated processing needed, from purely reactive (minimal processing) to strongly deliberative (requiring significant high-level cognition). This is similar to the response of biological organisms to damage, as was discussed.

Three different damage situations have been selected for discussion, two of them in considerable detail. These are (a) *critical damage*, defined as damage severe enough to threaten the integrity of the vehicle, generally caused by a sudden event (such as a major impact or explosion), (b) *sub-critical damage* which, although severe enough to require immediate action, does not invoke an emergency response, and (c) *non-critical* or *minor damage*, which does not necessarily require an immediate response, but which must be monitored continually in terms of its cumulative effects. Each of these will be described in terms of its components and analysed in terms of the two variables in the matrix, spatial extent and degree of deliberation. Since (a) is mostly reactive while (b) and (c) require varying degrees of deliberation, the latter two will be treated in sufficient detail to illustrate how the necessary computations may be accomplished in self-organising fashion in a multi-agent environment.

The following three sub-sections outline self-organizing responses to sub-critical and non-critical damage. This section, therefore, sets the context for these subsequent discussions.

Critical Damage

Critical damage means damage severe enough to threaten the integrity of the vehicle, requiring immediate action to ensure its survival. The initial response to such a situation must almost certainly be reactive, since the necessary actions will need to be implemented as rapidly as possible. It will almost certainly be global in extent, since the whole structure will need to know about events of this import, even if the initial reaction occurs in the neighbourhood of the damage site. In terms of the response matrix, a critical damage event may be represented as shown in Table 2.

In such situations time is of the essence, and both detection and communication must be done as quickly as possible. The requirements on the communications network are to send an alarm as rapidly as possible to one or more (probably) unknown locations

Table 2. Critical damage event response matrix

	Component	Spatial extent	Deliberation
	Detection	Local/Neighbourhood	Reactive
	Report	Neighbourhood or Global	Reactive
Actions:	Isolate damaged region	Global	Reactive

using only the adjacent-cell communication links. In addition, there may exist barriers to communication due to the network configuration itself or to significant prior and continuing damage. Thus the communications environment is largely unknown and changing, providing a major challenge. Some work has been done on these problems, but much more is needed (Li, Guo, & Poulton, 2004). Detection of critical damage by the cell network is also difficult because of the necessary time constraints, and it will almost certainly be good policy to err on the side of caution. Some kind of local activity measure will probably be best, but again very little research exists as yet.

The emergency response to critical damage will almost always be followed by more deliberative actions once the immediate safety of the vehicle has been assured. This is outlined in the response matrix, and follows a path very similar to that for sub-critical damage, which is described in the next sub-section.

Sub-Critical Damage

Sub-critical damage is taken to mean local damage to one or a number of cells which, although serious enough to require immediate action, does not threaten the immediate survival of the vehicle. In terms of the response matrix, such events may be broken down as described in Table 3.

Damage detection occurs at the local (cellular) level as described in the *Ageless Aerospace Vehicle* section, and is followed by a local response whose purpose is to define the extent of the damage and allow the assessment of its severity. For the AAV, this involves the self-organised formation of impact boundaries (Foreman, Prokopenko & Wang, 2003; Lovatt, Poulton, Price, Prokopenko, Valencia, & Wang, 2003; Prokopenko et al., 2005a; Wang & Prokopenko, 2004), which are described in some detail in the *Impact Boundaries* section below. When an assessment has been made, it must be communicated to some point on the vehicle from which appropriate action may be generated. This is not reactive, but may be at either neighbourhood or global level. The same issues apply regarding communications, which have been discussed in the *Critical Damage* sub-section.

The appropriate action will depend on circumstance, and three examples are given above. These range from local repair, which may indeed be reactive, to invoking a secondary inspection mechanism to obtain additional information about the nature and severity of the damage. This may be carried out within the neighbourhood, or it may require global interactions. An intriguing possibility for future aerospace vehicles is self-replication, where a replacement for the damaged section is manufactured by a self-organising process. This is discussed more fully in the *Shape Replication* section that follows.

Table 3. Sub-critical damage event response matrix

	Component	Spatial Extent	Deliberation
	Detection	Local	Reactive
	Local response	Neighbourhood	Moderately deliberative
	Assess	Neighbourhood	Moderately deliberative
	Report	Neighbourhood or Global	Strongly deliberative
Actions:	Secondary Inspection	Neighbourhood or Global	Strongly Deliberative
	Local repair	Local	Reactive or Moderately deliberative
	Self-replication	Global	Moderately deliberative

Non-Critical (Minor) Damage

The last damage scenario to be analysed in some depth is non-critical or minor damage. As outlined in the *Response Matrix* sub-section, such damage is typified by minor impacts, fatigue, or corrosion, processes which do not interfere immediately with the functioning of a cell, but which may lead to structural failure if accumulated over time. It is necessary to monitor such damage, not only to assess its long-term impact, but also because of its possible interaction with more serious types of damage. As would be expected considering the longer time-scale for this damage mechanism, assessment and action are quite deliberative, and of broader (neighbourhood or global) extent than previously discussed examples, although detection will most likely still be local. Referring to the response matrix, a possible breakdown for non-critical damage is as described in Table 4.

Although the detection of non-critical damage is local and reactive, all other steps are either broader in extent or degree of deliberation. This is to be expected since any assessment and resulting action must involve a number of non-critical damage sites. There is thus a need for the self-organisation of information regarding non-critical damage so that the relevant assessment can be made and acted upon. The most important of such information comprises the locations of damage sites and the severity of their damage, and there are several ways in which this information may be made to self-organise. One promising method is by the formation of an impact network, which is essentially a way of advantageously linking non-critical damage sites (Prokopenko et al., 2005a; Wang, Valencia, Prokopenko, Price, & Poulton, 2003). Not only does this make the necessary information available for processing, but it offers a mechanism by which secondary inspection (or repair) agents may rapidly assess the damage. The formation and use of impact networks is discussed in the section on *Impact Networks and Ant Colonies*.

IMPACT BOUNDARIES

Typically, the damage on the AAV skin caused by a high-velocity impact is most severe at the point of impact (an epicentre). It will be assumed that not only are the cells

Table 4. Non-critical damage event response matrix

	Component	Spatial extent	Deliberation
	Detection	Local	Reactive
	Local assessment	Local	Moderately deliberative
	Form impact network	Global	Moderately deliberative
	Report network status	Neighbourhood or Global	Moderately deliberative
	Assess network status	Global	Strongly deliberative
Actions:	Secondary inspection	Global	Strongly deliberative
	Repair	Global	Strongly deliberative

at the epicentre severely damaged, but that damage spreads to neighbouring cells, perhaps as a result of severe vibration, blast, or electromagnetic effects. One effect of this propagated damage is likely to be observed as damage to the communication capability of the neighbouring cells. For the sake of a specific example, we simulate the effect of this extended damage by assuming a communications error rate that is propagated out with an exponential decay to a certain radius (Lovatt et al., 2003). In this example, the damage can be characterised by a probability P_i of an error corrupting a message bit i, dependent on proximity of the affected communication port to the epicentre:

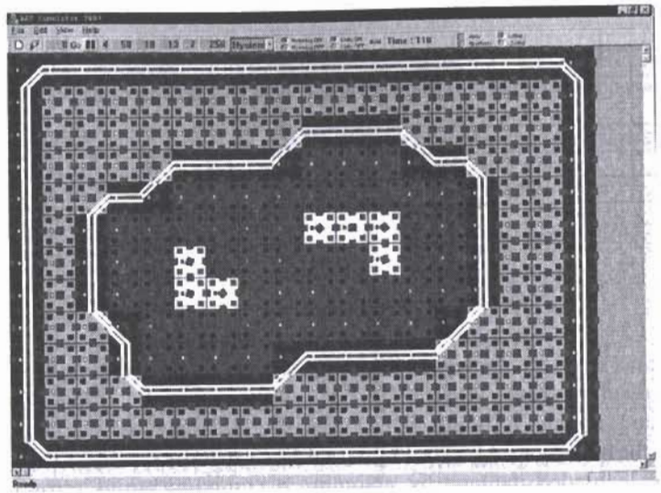
$$P_i = \frac{1}{2}(1 - \frac{d}{R})^{\alpha} \qquad (1)$$

where d is the distance between the involved communication port and the epicentre of the impact with the radius R, and α is the exponential decay of the communication loss (we have investigated a range of values, including linear decays, $\alpha = 1$, and high-order polynomial decays, $\alpha \leq 7$). Multiple impacts result in overlapping damaged regions with quite complex shapes.

In this section we describe multi-cellular *impact boundaries*, self-organising in the presence of cell failures and connectivity disruptions, and their use in damage evaluation and possibly self-repair.

It is desirable that an impact boundary, enclosing damaged areas, forms a continuously- connected closed circuit. On the one hand, this circuit may serve as a reliable communication pathway around the impact-surrounding region within which communications are compromised. Every cell on a continuously-connected closed circuit must always have two and only two neighbour cells, designated as the circuit members (circuit-neighbours of this cell). On the other hand, a continuously-connected closed impact boundary provides a template for repair of the impact-surrounding region, uniquely describing its shape (Figure 5). Both these functionalities of impact boundaries can be contrasted with non-continuous "guard walls" investigated by Durbeck and Macias (2002) that simply isolate faulty regions of the Cell Matrix, without connecting elements of a "guard wall" in a circuit. An impact boundary enables a shape-replication of a multi-

Figure 5. White cells are destroyed, red (dark-grey) cells form "scaffolding", and blue (black) cells form the "frame" (boundary links are shown as white double-lines).



cellular impact-surrounding region, which can serve as an example of an end-to-end solution to an important SHM sub-task of damage evaluation and subsequent self-repair. The damage evaluation part (emergent evolvable impact boundaries) is implemented in the AAV-CD, while the shape-replication part is only simulated at this stage.

In order to serve either as a communication pathway or a repair template, an impact boundary should be *stable* despite communication failures caused by proximity to the epicentre, and such stability is our primary aim. In pursuing this aim, we deal with the following spatial self-organising layers:

- **Scaffolding** region, containing the cells that suffered significant communication damage;
- **Frame boundary**, an inner layer of normal cells that are able to communicate reliably among themselves; and
- **Closed impact boundary**, connecting the cells on the frame boundary into a continuous closed circuit by identifying their circuit-neighbours.

The "frame" separates the scaffolding region from the cells that are able to communicate to their normal functional capacity. In order to support a closed continuously-connected circuit, a *regular* frame should not be too thin (a scaffolding cell must not be adjacent to a normal cell), and should not be too thick (there must be no frame cells in the direction orthogonal to a local frame fragment). These internal layers (scaffolding, frame, and closed boundary) completely define an impact-surrounding region as a layered spatial hierarchy. In general, the impact-surrounding region can be seen as an example of annular spatial sorting: "forming a cluster of one class of objects and

surrounding it with annular bands of the other classes, each band containing objects of only one type" (Holland & Melhuish, 1999). It could be argued that, as an emergent structure, the impact-surrounding region has unique higher-order properties, such as having *an inside* and *an outside* (Prokopenko et al., 2005a).

Evolvable Localised Algorithm

The algorithm producing continuously-connected closed impact boundaries and the metrics quantitatively measuring their spatiotemporal stability are described in Foreman, Prokopenko, and Wang (2003) and Prokopenko et al. (2005a), while a genetic algorithm evolving agent properties which form impact boundaries satisfying these spatiotemporal metrics is presented in Wang and Prokopenko (2004). Here we briefly sketch the main elements of the evolved algorithm.

Every cell sends a *Ping* message to each of its neighbours regularly, and an *Acknowledgment* reply when it receives a Ping message. For each communication port i, a binary circular array A_i is used to store the communication histories for acknowledgments. The size of the array is called the communication history length μ. For each communication port, a Boolean success variable P_i is set to true if the percentage of *Acknowledgments* received in the A_i is greater than or equal to a certain threshold P. This variable is hysteretic: it changes only when a sufficient communication history is accumulated. This lagging of an effect behind its cause provides a temporary resistance to change and ensures a degree of stability in the treatment of communication connections between any two cells. A neighbour i is considered to be *communicating* when P_i is true. The algorithm uses the following main rules:

- Each cell switches to the *Scaffolding* state and stops transmitting messages if the number of communicating neighbours ν is less than a certain threshold $\mathcal{N}$. For example, if $\mathcal{N} = 1$, then a cell switches to *Scaffolding* state if there are no communicating neighbours ($\nu < 1$).
- Each cell switches to *Frame* boundary state S_f if there is at least one communicating neighbour and at least one miscommunicating neighbour.
- Each cell switches to *Closed* boundary state S_c if the cell state is S_f, and there are at least two *communicating* neighbours.

The *Closed* boundary cells send and propagate (within a *time to live* period τ) specific *Connect* messages, leading to self-organisation of a continuous impact boundary. The cells that stopped transmitting messages may need to resume communications under certain conditions, for example, when a repair action is initiated, and their neighbours are again ready to receive communications (that is, when the cause of asymmetry is eliminated). The conditions for resumption of communications have to be precise so that they are not reacted upon prematurely, interfering with boundary formation. A variant of these *recovery* conditions is given:

- Each cell switches to the *Recovery* state S_r if the sequence $P_0, \ldots, P_3$ describing the states of all four ports does not change for a specified number of consecutive cycles π_1.
- A cell stays in the *Recovery* state S_r and may send communication messages during the next π_2 cycles.

In general, the described policy achieves the desired stability and continuity of self-organising impact boundaries. In addition, we have observed emergent spatiotemporal structures — recovery membranes — that separate the boundaries from recovering cells. A recovery membrane always forms on the inside of the closed boundary, and on the outside of the recovering area. Interestingly, unlike scaffolding and frame boundary, the membrane is not a designated state into which a cell can switch. Membrane cells shut down their communications like other scaffolding cells, but do not resume communications because *recovery* conditions are not applicable, as the miscommunicating neighbours are not stable. Without a membrane, the cells on the frame boundary would be confused by intermittent messages from scaffolding cells attempting recovery. Figure 6 illustrates a checkered-pattern recovery membrane shown with dark-grey colour, while the recovering cells are shown in yellow (darker shade of white).

The threshold $\mathcal{N}$, limiting the number of communicating neighbours in switching to the *Scaffolding* state, significantly affects smoothness of a resulting boundary. In particular, if $\mathcal{N} = 1$ — in other words, a cell switches to the *Scaffolding* state if there are no *communicating* neighbours ($\nu < 1$) — then some boundary links may not be "smooth": There are more than two ports connected by the link (Figure 7). If $\mathcal{N} = 2$ — that is, a cell switches to the *Scaffolding* state if there is at most one *communicating* neighbour ($\nu < 2$) — then all boundary links are "smooth" (for instance, the case depicted in Figure 6). If $\mathcal{N} = 3$, then any impact boundary is a rectangle. Finally, if $\mathcal{N} = 4$, then the impact-surrounding region fills the whole of the AAV array. This simple taxonomy of boundary types will be useful when we classify shape-replication algorithms as well.

Figure 6. Five white cells at the epicentre are destroyed, scaffolding cells that attempt recovery are shown in yellow (darker shade of white); a recovery membrane, shown in red (dark-grey), "absorbs" and separates them from the frame, shown in blue (black)

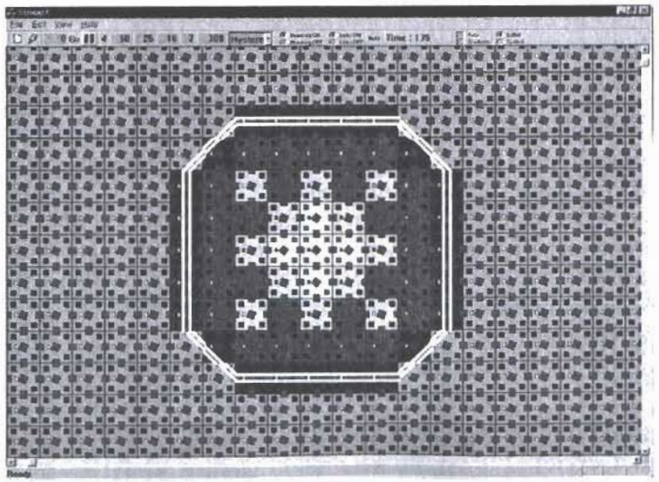
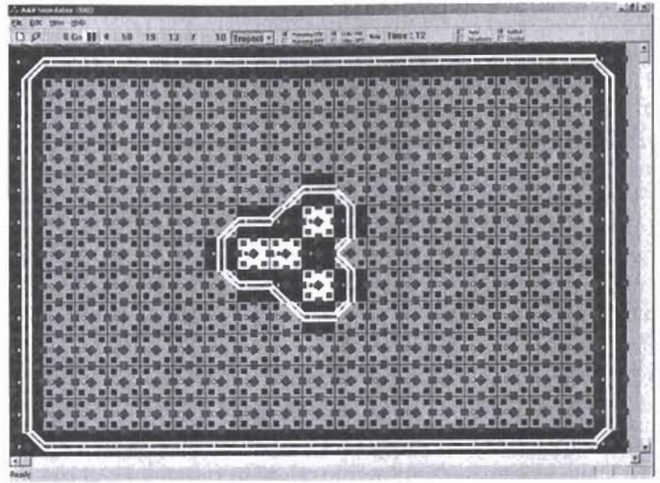


Figure 7. Four white cells are destroyed (a recovery membrane is not shown); the middle boundary link on the right-hand side is not smooth.



Spatiotemporal Stability, Phase Transitions, and Evolving Boundaries

The evolution of impact boundaries is based on spatiotemporal metrics incorporated within a fitness (objective) function. The analysis presented by Foreman, Prokopenko, and Wang (2003) and Prokopenko et al. (2005a) used two metrics to characterise stability of emergent impact boundaries: spatial and temporal.

The spatial metric is based on the variance in the size of the connected boundary-fragment (CBF). A CBF is simply a set F of cells in the *Closed* state S_c such that every cell in F is connected with at least one other cell in F, and there exists no cell outside F which is connected to at least one cell in F (an analogue of a maximally-connected subgraph or a graph component). We calculate the maximum size $H_{sp}(t)$ of CBFs in self-organising impact boundaries at each cycle. Its variance σ_{sp}^2 over time is then used as a spatial metric within the objective function. This metric is inspired by random graph theory and is intended to capture *spatial connectivity* in impact boundaries. A continuous boundary may, however, change its shape over time, without breaking into fragments, while keeping the size of CBF almost constant. Therefore, a temporal metric may be required as well.

In order to analyse *temporal persistence*, we consider state changes in each cell at every time step. Given six symmetric boundary links possible in each square cell ("left-right", "top-bottom", "left-top", and so on), there are 2^6 possible boundary states (including "no-boundary"), and $m = 2^{12}$ transitions. The entropy $H_{temp}(t)$ of a particular frequency distribution $S_i(t)$, where t is a time step, and i is a cell transition index: $1 \leq i \leq m$, can be calculated as follows (Equation 2):

$$H_{temp}(t) = -\sum_{i=1}^{m} \frac{S_i(t)}{n} \log \frac{S_i(t)}{n} \qquad (2)$$

where n is the total number of cells, and $S_i(t)$ is the number of times the transition i was used at time t across all cells. The variance σ_{temp}^2 of the entropy $H_{temp}(t)$ over time is used as a temporal metric within the objective function.

Our task is complicated by the fact that emergent structures are characterised by a *phase transition* detectable by either σ_{sp}^2 or σ_{temp}^2, rather than a particular value range. Therefore, simply rewarding low values for these entropy-based metrics would be insufficient. In particular, it has been observed (Foreman, Prokopenko, & Wang, 2003; Wang & Prokopenko, 2004) that both metrics are low-to-medium for algorithms with zero-length communication μ (tropistic algorithms and chaotic regimes — Figure 8), increase dramatically for μ in the range $1 \leq \mu \leq \mu_0$, where μ_0 is a critical value at and below which complex unstable behaviours occur (Figure 9), and undergo a phase transition to very low values when $\mu > \mu_0$ (ordered regimes).

The critical value μ_0 is, of course, dependent on all other parameters used by the algorithm. Nevertheless, the chaotic regimes, which are more stable simply due to a small number of connections, can often be identified by a low average $\overline{H_{sp}}$ of the maximum sizes $H_{sp}(t)$ of CBFs in impact boundaries, ruling out at least zero-length histories. In particular,

Figure 8. A chaotic boundary with $\overline{H_{sp}} \leq 16$ and zero-length communication μ; a membrane does not form at all (both σ_{sp}^2 and σ_{temp}^2 are low-to-medium)

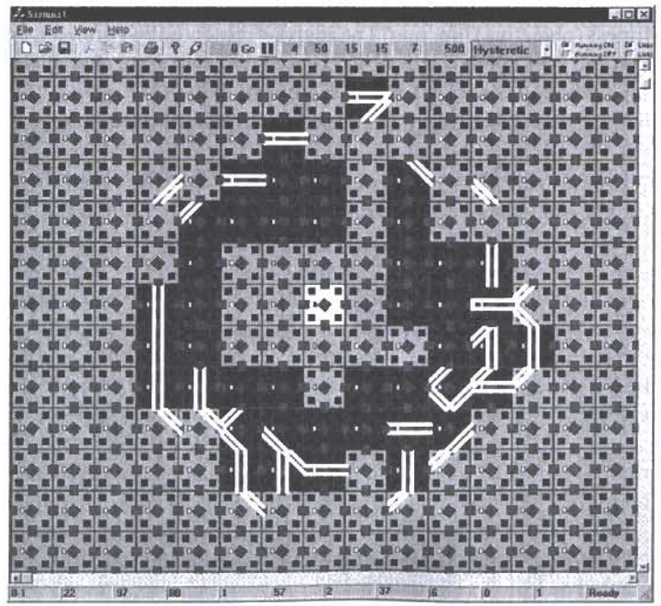
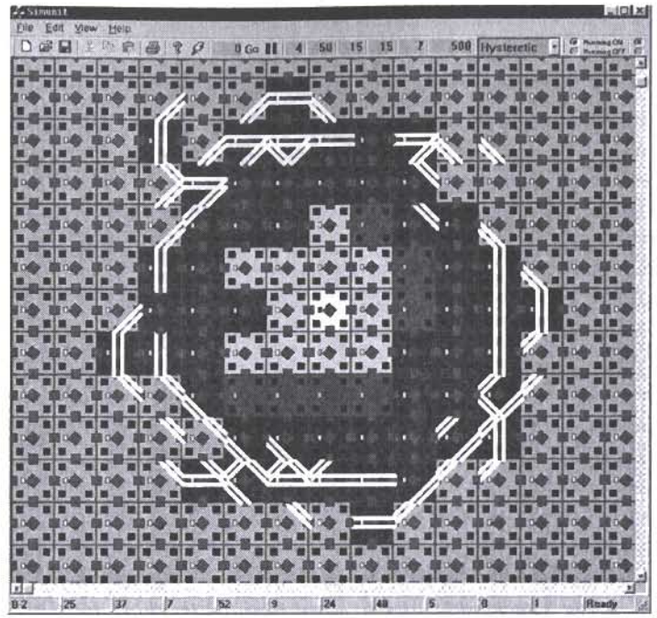


Figure 9. An unstable boundary with μ close to its critical value; the membrane is fragmentary (both σ_{sp}^2 and σ_{temp}^2 are close to their peaks; i.e. a phase transition)



impact boundaries with the average $\overline{H_{sp}} \leq 16$ can be safely ruled out; the resulting chaotic patterns, illustrated in Figure 8, are of no interest.

On the other hand, a preference among ordered regimes towards shorter histories is another useful identifier of a phase transition and the critical value μ_0. Besides, a shorter communication history μ enables a quicker response, as do lower values of τ and π_1.

Thus, our experiments used minimisation of the following objective function:

$$f(\beta) = \begin{cases} M & \text{if } \overline{H_{sp}} \leq 16; \\ \frac{1}{2}(4.0\sigma_{sp}^2 + 10^5 \sigma_{temp}^2) + \mu + \tau + \pi + \beta \overline{H_{sp}} & \text{if } \overline{H_{sp}} > 16 \end{cases} \qquad (3)$$

where M is the maximal integer value provided by the compiler. The coefficient β reflects the relative importance of the length of impact boundaries in the objective function; sometimes it may be as important to obtain the smallest possible impact perimeter as it is to maintain the shortest possible communication history. We alternated between $\beta_1 = 0.25$ and $\beta_2 = 2.0$.

Each experiment involves an impact at a pre-defined cell and lasts 500 cycles; the first 30 cycles are excluded from the series $H_{sp}(t)$ and $H_{temp}(t)$ in order not to penalise

longer history lengths μ. We repeat the experiment three times for every chromosome (a combination of parameters) and average the objective (fitness) values obtained over these runs. The details of the genetic operators, the employed replacement strategy, and a comparative analysis of metrics are described in Wang and Prokopenko (2004). Here we only summarise the results.

The experiment minimising the objective function $f(0.25)$ evolved solutions with long robust and continuous impact boundaries with $\overline{H_{sp}} = 40$ (Figure 10), around large impact-surrounding regions, while requiring fairly short hysteresis: $\rho = 2$ and $\pi = 8$. The stabilisation of an impact boundary around a large region occurs at the periphery of the communication damage, where the communication failure probability falls to zero due to the error correction code, and the process has a cascading nature, where the boundary expands to eventually cover the entire impact-surrounding region. In summary, the case $\beta = 0.25$ results in longer boundaries that are sometimes capable of morphing without breaking into fragments.

On the other hand, minimisation of $f_{sp}(2.0)$ resulted in more compact impact-surrounding regions ($\overline{H_{sp}} = 32$, Figure 11) and thinner membranes, at the expense of longer hysteresis: $\rho = 6$ and $\pi = 8$. These boundaries generally keep the shape of a regular octagon. This case (b = 2.0) results in shorter boundaries that cannot morph without breaking into fragments, so any instability leads to fragmentation. Both solutions favoured $\tau = 1$ as expected for square cells.

Figure 10. A large checkered-pattern membrane, with short hysteresis, within a morphing but closed and continuous boundary ($\beta = 0.25$)

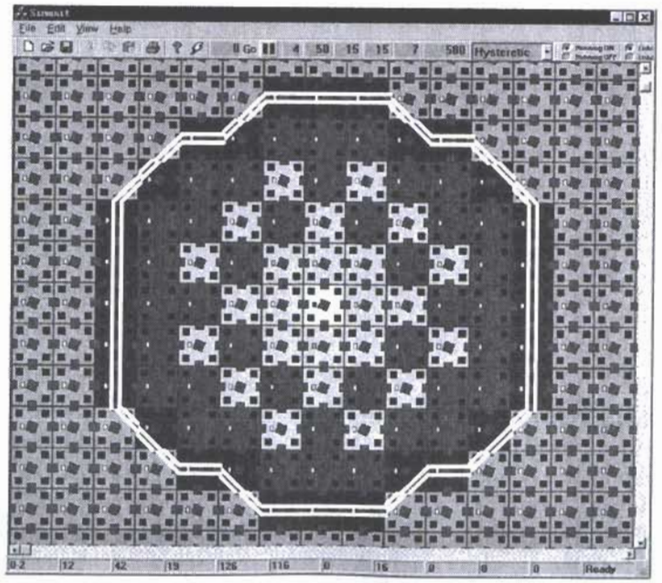
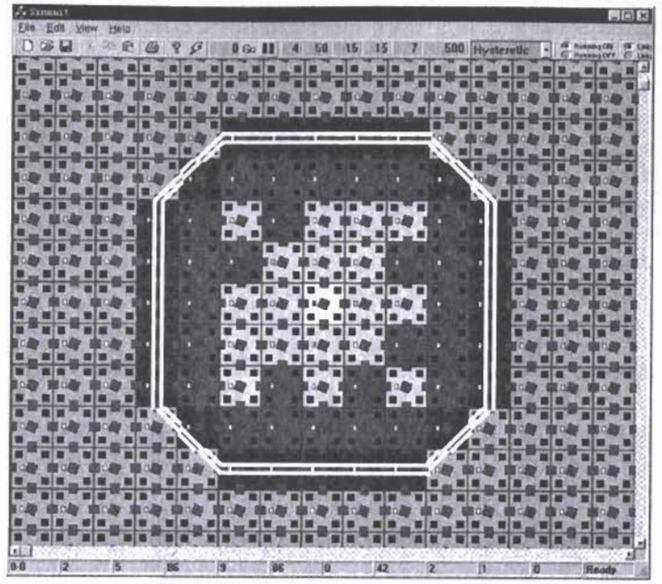


Figure 11. A small membrane, with long hysteresis, within a regular octagonal boundary ($\beta = 2.0$)



These results are promising and demonstrate the possibility for a multi-objective design of localised algorithms. In particular, the desired response time as well as size (and potentially, shape) of impact boundaries become the *design parameters* and may be specified in advance, leaving the precise logic and parameterisation of the localised algorithms to selection pressures. We believe that the proposed methodology is well suited to the *design at the edge of chaos*, where the design objective (for example, a specific shape) may be unstable, while other parameters (such as the response time) may be optimal.

The impact boundaries form patterns that may be used in damage assessment and diagnostics, as well as templates for repair, and provide reliable communication pathways around impact-surrounding regions. Their multiple roles illustrate two kinds of emergence: *pattern formation* and *intrinsic emergence*, distinguished by Crutchfield (1994):

- **Pattern formation** refers to an external observer who is able to recognise how certain unexpected features (patterns) "emerge" or "self-organise" during a process (for example, convective rolls in fluid flow, and spiral waves and Turing patterns in oscillating chemical reactions). The patterns may not necessarily have specific meaning ***within the system***, but obtain a special meaning *to the observer* when detected;

- **Intrinsic emergence** refers to the emergent features which are important *within the system* because they confer additional functionality to the system itself, like supporting global coordination and computation (for instance, the emergence of coordinated behaviour in a flock of birds allows efficient global information processing through local interaction, which benefits individual agents).

In the next section we shall illustrate how stable and continuous impact boundaries can be used as templates for multi-cellular shape-replication, while the following section will describe a self-organising communication mechanism among remote cells. This mechanism may be used, in particular, to communicate the information represented by the emergent boundary patterns to remote cells playing the role of observers and/or controllers, if necessary.

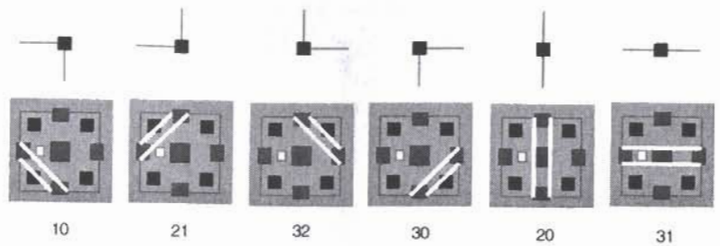
SHAPE REPLICATION: TOWARDS SELF-REPAIR

In general, individual failed cells can be replaced one by one — this is, after all, the point of having a scalable solution. Moreover, any impact-surrounding region enclosed within its impact boundary can also be repaired by replacing individual failed cells one by one. Sometimes, however, it may be required to replace an impact-surrounding region in one step, for example, to minimise the overhead of disconnecting individual cell-to-cell links. Replacing the whole region within a boundary would require a removal of only the links between the boundary and normal cells. In this sub-section, we provide an example where a self-organised impact boundary (produced by the evolved algorithm) may be used in self-repair, or more precisely, in shape replication.

Given the planar grid topology, each cell on the closed impact boundary may have six boundary links, connecting ports "left-right", "left-top", and so on. Enumerating four communication ports from zero to three ("bottom" to "right" clockwise) allows us to uniquely label each boundary link with a two-digit number λ — for example, "32" would encode a link between the "right" and "top" ports (Figure 12). Then, the whole impact boundary can be encoded in an ordered list of these labels. However, in order to replicate the bounded shape, filling it cell by cell, we need to introduce a system of coordinates relative to a cell containing the shape list. More precisely, the boundary genome is a list of triples (α,β,λ), where (α,β) are relative coordinates of a cell with the boundary link λ.

The shape replication algorithms developed in the context of AAV (Prokopenko & Wang, 2004) are based on the principles of multi-cellular organisation, cellular differentiation, and cellular division — similar to the embryonics approach (Mange, Sanchez, Stauffer, Tempesti, Marchal, & Piguet, 1998; Sipper, Mange, & Stauffer, 1997). A desired shape is encoded when an emergent impact boundary inspects itself and stores the "genome" in a "mother" cell. The genome contains both data describing the boundary and a program of how to interpret these data. The mother cell is then seeded in a new place outside the affected AAV array. An execution of its program initiates *cell-replication* in the directions encoded in the genome (Figure 13). Each cell-replication step involves copying of the genome (both data and the program) followed by differentiation of the data: an appropriate shift of certain coordinates. Newly produced cells are capable of

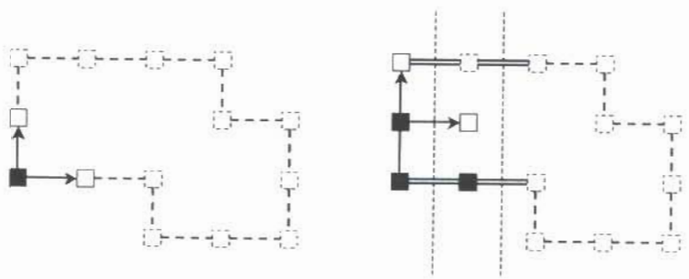
Figure 12. Boundary links



cellular division, continuing the process until the encoded shape is constructed (Figure 14).

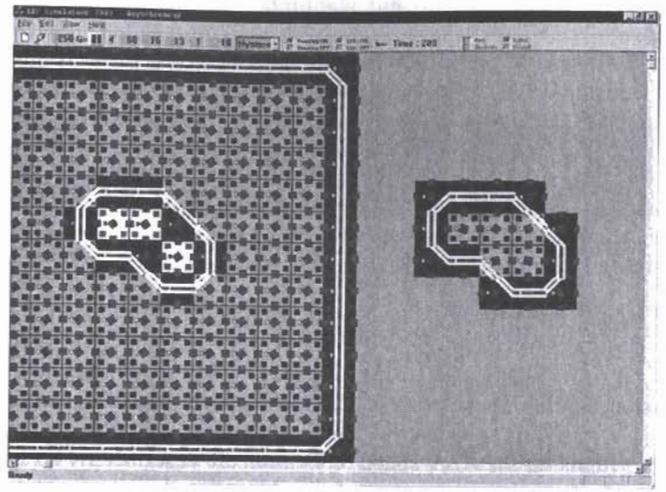
Two algorithms for the AAV shape replication are described by Prokopenko and Wang (2004). The first algorithm solves the problem for connected and disconnected shapes. The second algorithm, in addition, recovers from possible errors in the "genome", approximating missing fragments. In particular, the genome is partially repaired (Figure 15) within each cell which detected a discontinuity. Although the repaired genome does not cover all the missing cells, it does not introduce any cells which were not in the original shape, exhibiting the soundness but not completeness property. In other words, the repaired boundary is contained within the original shape. Importantly, there is a redundancy in the shape replication process: other cells which did not suffer any damage would successfully replicate the parts not encoded in the partially repaired genomes.

Figure 13. Shape replication: Boundary cells encoded in the genome but not yet produced are shown with dashed lines.



Left: A black cell (seed) produces two white cells, indicated by arrows. Right: Two more cells are being produced: one of them is a scaffolding cell, pointed to by the horizontal arrow (the inside direction is recognised by the vertical strip being 'crossed' above and below the considered location).

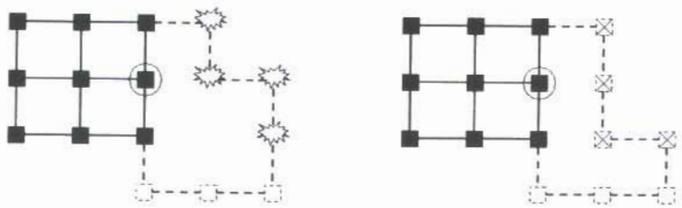
Figure 14. Completed shape replication



The shape-replication algorithms handle both standard ("blueprint") and non-standard emergent shapes, self-organising in response to damage. Moreover, it is possible to combine these types. For example, structural data can be encoded in the form of triples, and a given genome can be extended in run-time with the data produced by self-inspecting emergent boundaries. Similarly, the self-repair phase within a cell which detected an anomaly in the genome may draw some data from the structural "blueprints" rather than approximate segments between disconnected fragments.

Importantly, the first algorithm, not involving a recovery of the genome from possible errors, replicates shapes encoded in either smooth or non-smooth boundaries; it does not depend on the threshold $\mathcal{N}$, limiting the number of communicating neighbours in switching to the scaffolding state. The second, genome-repairing, algorithm, however,

Figure 15. The cell shown inside a circle attempts self-repair



Left: the corrupted triples are shown with the 'star'-like signs. Right: the repaired triples are marked with crosses.

cannot deal with non-smooth boundaries — that is, when $\mathcal{N} = 1$ and a cell switches to the *scaffolding* state if there are no communicating neighbours. Thus, adding the selection force rewarding genome-recoverability would lead to evolution of only smooth and stable impact boundaries ($\mathcal{N} = 2$). In other words, the taxonomy of boundary types, based on the threshold $\mathcal{N}$, is related to a classification of shape-replication algorithms; for example, it is conceivable that some replication sub-tasks may tolerate only rectangular shapes ($\mathcal{N} = 3$).

The shape replication process described in this section can be used in repairing the impact-surrounding regions in one step. As mentioned before, this is not the only feasible strategy, and failed cells can be replaced individually. In addition, there is a possibility to employ self-healing materials; however, this reaches beyond the scope of our investigation.

IMPACT NETWORKS AND ANT COLONIES

Decentralised inspection across the AAV network array may require an *impact network* among cells that registered impacts with energies within a certain band (for example, non-critical impacts). The self-organising impact networks create an adaptive topology allowing inspection agents (communication packets or, potentially, swarming robots) to quickly explore the area and evaluate the damage (for example, identify densities of impacts typical for a meteor shower, evaluate progression of corrosion, or to trace cracks propagation) — particularly where a number of individually non-critical damage sites may collectively lead to a more serious problem. Robotic agents may need an impact network which solves a travelling salesperson problem (TSP). On the other hand, a shortest or *minimum spanning tree* (MST) is often required in order to enable decentralised inspections when virtual (software) agents are employed, and may provide a useful input for the TSP. In this section we present an extension of an ant colony optimisation (ACO) algorithm, using an adaptive dead reckoning scheme (ADRS) and producing robust and reconfigurable minimum spanning trees connecting autonomous AAV cells. A novel heuristic is introduced to solve the blocking problem: reconfiguration of an existing path which is no longer available or optimal. Dynamic formation of a robust reconfigurable network connecting remote AAV cells that belong to a specific class was analysed in our previous work (Abbott et al., 2003; Wang et al., 2003). The ACO algorithm developed in these studies successfully approximates minimum spanning trees, but occasional alternative paths around critically damaged areas may still emerge, competing with the shortest paths and slowing the algorithm's convergence.

Let us define an AAV impact network. A two-dimensional AAV array can be represented by a planar grid graph $G(V,E)$: the product of path graphs on m and n vertices, which are points on the planar integer lattice, connected by the edges $E(G)$ at unit distances (Figure 16). The cells which represent specific points of interest (for example, the cells which detected non-critical impacts, or the cells playing a role of local "hierarchs", "observers", or "controllers") form a sub-set P of $V(G)$. We need to identify those edges Z in $E(G)$ which connect the vertices in P minimally, so that the total distance (a sum of unit distances assigned to edges Z) is shortest. This problem is essentially the standard minimum spanning tree problem, except that a spanning tree is defined for a

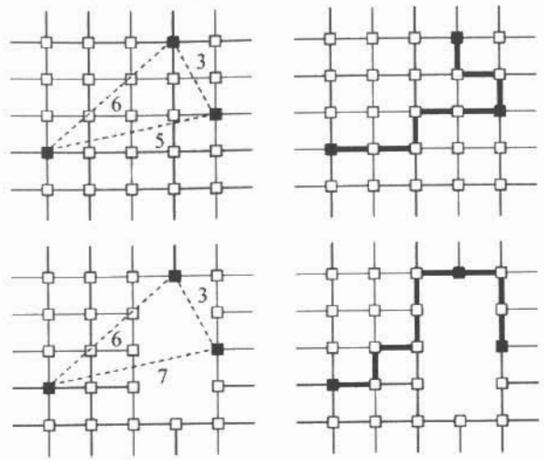
graph, and not for a set of vertices. Our problem is sometimes referred to in the literature as the *rectilinear minimum (terminal-) spanning tree* (RMST) problem — a fundamental problem in VLSI design — and in which the vertices in P are referred to as *terminals* (Kahng & Robins, 1995). The important difference between MST and RMST is that rather than choosing MST edges out of the graph edges $E(G)$ directly connecting pairs of vertices, we need to find multi-edge rectilinear paths between vertices in P, minimising the total distance. This can be done via an auxiliary complete graph A, whose vertex set is P and in which the edge pq for $p, q \in P$ with $p \neq q$ has length equal to the Manhattan distance between nodes p and q. After a standard MST A_m is identified in the graph A, we merely need to convert all edges in A_m to rectilinear paths on the grid graph G (Figure 16).

However, the impact network problem, however, is complicated by possible "obstacles" created by discontinuities in the AAV grid graph G. Initially, the grid graph G is *solid*; it does not have any "holes", so its complement in the infinite orthogonal planar grid is connected. New critical impacts may create such holes in the grid. Figure 16 illustrates the RMST problem with two scenarios. The first case is shown in the top part, and involves three edges and a simple MST A_m with the total distance of eight. The second case is shown in the bottom part: some cells are destroyed (the corresponding vertices are removed), and the auxiliary complete graph should be updated because one shortest path has changed (from five to seven). This requires a recomputation of its MST (the new MST distance is nine), with another edge being selected and converted to a rectilinear path. This illustrates that a new obstacle may not just require that a new shortest path is found between the two involved cells (the problem investigated by Wu, Widmayer, Schlag, & Wong, (1987)), but rather than the whole MST is re-evaluated. Moreover, there are cases when a cell/terminal is no longer needed to be included in the RMST, or a new cell/terminal needs to be added. Incremental updates of an old rectilinear spanning tree may provide a practical solution, but a quick divergence from an RMST is a significant problem.

Thus, from a graph-theoretic standpoint, the representation of the impact network problem changes over time due to insertion of new nodes (for example, non-critical impacts) or deletion of old nodes no longer fitting the impact range, while the problem's properties change due to varying connection costs (for example, critical impacts destroying existing paths). In short, the problem changes concurrently with the problem-solving process (Prokopenko et al., 2005a), and we need a dynamic and decentralised computation of a rectilinear minimum terminal-spanning tree in the presence of obstacles. If the information (such as the auxiliary graph A) was available in one central point, then the RMST problem would essentially become an MST problem, with a subsequent conversion to rectilinear paths. In this case the required computation itself would not be NP-hard, although the fully dynamic case, in which both insertions and deletions must be handled online, without knowing the sequence of events in advance, would still be quite intensive. Eppstein (1996) estimated a running time of a fully dynamic graph minimum spanning tree algorithm as $O(n^{1/2}\log^2 n + n^{\varepsilon})$, where ε is a (very small) constant, *per update*. In our case, the auxiliary graph A is not even known at any single node/cell, so the desired algorithm should be both *decentralised* and *fully dynamic*.

These factors suggested that the problem of forming minimum spanning trees on the AAV skin can be efficiently tackled by ant colony optimisation (ACO) algorithms,

Figure 16. Three impact nodes are shown in black



The top-left figure shows a complete auxiliary graph A (dashed lines) with three edges. The conversion of its MST to rectilinear paths on the AAV grid graph is shown in the top-right figure (bold edges). Two lower figures show the graph with some vertices removed. The bottom-left figure shows an updated auxiliary graph A, and the bottom-right figure shows conversion of the new MST to rectilinear AAV paths.

proposed and enhanced over recent years by Dorigo and his colleagues (Colorni, Dorigo, & Maniezzo, 1992; Dorigo & Di Caro, 1999; Dorigo, Maniezzo, & Colorni, 1996), rather than distributed dynamic programming (Bellman-Ford) algorithms. Essentially, the ACO algorithms use the ability of agents to indirectly interact through changes in their environment (*stigmergy*) by depositing pheromones and forming a pheromone trail. They also employ a form of *autocatalytic* behaviour — *allelomimesis*: the probability with which an ant chooses a trail increases with the number of ants that chose the same path in the past. The process is thus characterised by a positive feedback loop (Dorigo, Maniezzo, & Colorni, 1996). An overview of the ACO meta-heuristic and its applicability can be found in Dorigo and Di Caro (1999).

In the AAV-CD the ants are implemented as communication packets, so the policies are implemented via appropriate message passing, where the cells are responsible for unpacking the packets, interpreting them, and sending updated packets further if necessary. Thus, ants cannot move into the cells with damaged (or shutdown) communication links, so critically-impacted cells form obstacles, and the ants are supposed to find the shortest paths around them using positively-reinforced pheromone trails. For our problem, it is impractical to use two types of pheromone (such as "nest" and "food") because each impact cell (node) serves both as a "nest" and a "food" source. Therefore, having two types of pheromone per node would have created multiple pheromone fields, combinatorially complicating the network. In addition, dissipation of pheromone over

large distances is not practical either, as it would lead to "flooding" of the network with messages. Hence, the algorithms developed for the AAV network use only one type of non-dissipative evaporating pheromone.

The ACO-ADRS Algorithm

The algorithm presented in Abbott et al. (2003) and Wang et al. (2003) was based on a hybrid method of establishing impact networks, using a single impact gradient field (IGF) and a dead reckoning scheme (DRS), complementing the autocatalytic process of ant-like agents. Following Prokopenko et al. (2005a) and Prokopenko, Wang, and Price (2005), we summarise here a major variant of this algorithm, without an IGF, and relying only on DRS. The behaviour of exploring ants includes the following:

(E1) each impact node generates a number of exploring ants every T cycles; each ant has a "time to live" counter t_k, decremented every cycle;

(E2) an exploring ant performs a random walk until either (a) another impact node is found, or (b) the ant has returned to the home impact node, or (c) the ant can move to a cell with a non-zero trail intensity;

(E3) if an exploring ant can move to a cell with a non-zero trail intensity, the destination cell is selected according to transitional probabilities;

(E4) at each step from cell i to cell j, an exploring ant updates the x- and y-shift coordinates from the home node (initially set to 0).

The DRS requires that each ant remembers the x- and y-shift coordinates from the home node. These coordinates are relative, they simply reflect how many cells separate the ant from the home node in terms of x and y at the moment, and should not be confused with a "tabu" list of an ACO agent containing all visited nodes in terms of some absolute coordinate or identification system. The DRS enables the agents to head home when another impact node is located:

(R1) when another impact node is found, the exploring ant switches to a return state, remembers the ratio g = y/x corresponding to the found node's coordinates relative to the home node, and starts moving back to the home node by moving to cells where the y- and/or x-shift coordinates(s) would be smaller and their ratio would be as close as possible to g; if both x- and y-shift are zero (the home node), the returning ant stops;

(R2) if the cell suggested by the DRS (minimisation of x- and/or y-shift, while maintaining g) cannot be reached because of a communication failure (an obstacle), the ant selects an obstacle-avoiding move according to the transitional probabilities; upon this selection the ant keeps to the chosen path until the obstacle is avoided, as recognised by comparison of current y/x ratio with g;

(R3) each cycle, a returning ant deposits pheromone in the quantity inversely proportional to the traversed return distance q (q is incremented by 1 each cycle); the deposited pheromone is limited by a pre-defined maximum φ_{max}.

The pheromone is deposited on the cells themselves rather than communication links; we deal with pheromone trail intensities φ_j at the cell j, used in calculating transitional probabilities and determining which neighbour cell should be chosen by an incoming ant packet to continue their travel. The intensity of trail $\varphi_j(t)$ on the node j gives

information on how many ants have traversed the node in the past, and is updated each time an ant agent k passes through the node:

$$\varphi_j(t) = \min(\varphi_j(t) + \frac{\sigma_k}{q_k(t)}, \varphi_{max}) \tag{4}$$

where σ_k is a constant quantity specified for each generated ant k, q_k is the distance traversed by the ant k, and φ_{max} is a limit on pheromone trail intensity. Intuitively, the quantity σ_k represents a pheromone reserve of the ant k, consumed during the return trip. At the beginning of each cycle, the pheromone evaporates at the rate $\rho \in (0,1)$:

$$\varphi_j(t) = (1-\rho)\,\varphi_j(t) = \psi\,\varphi_j(t) \tag{5}$$

where ψ is the retention rate. A study of the impact network stability is provided by (Prokopenko et al., 2005a). An improvement to the DRS algorithm included *adaptive* pheromone reserve quantity σ_k and "time to live" counter τ_k, and a "pause" heuristic (Prokopenko, Wang, Scott, Gerasimov, Hoschke, & Price, 2005b). The pheromone reserve is adaptively allocated by the generating node, based on the ants returned to the node in the past:

$$\sigma_k = \max(\gamma_1 \hat{q}, \sigma_{min}) \tag{6}$$

where $\hat{q}$ is the minimal distance traversed by the returned ants, γ_1 is a scaling factor, and σ_{min} is a lower limit for the pheromone reserve allocated for an ant. Analogously,

$$\tau_k = \min(\gamma_2 \hat{q}, \tau_{max}) \tag{7}$$

where τ_{max} is an upper limit for the counter, and γ_2 is a scaling factor. Equations (6) and (7) define the adaptive dead reckoning scheme (ADRS), which contributes to a faster reconfiguration of trails and minimum spanning trees.

The "pause" heuristic contributes to a better convergence of the DRS and ADRS algorithms. Let us consider decisions of a returning ant in the situation when an obstacle blocks a DRS path towards the home node. If the ants used both "nest" and "food" pheromones, then an ant returning to the "nest" and tracing the "nest" pheromone (while depositing the "food" pheromone) would benefit from the stigmergy as both shorter and longer paths around the obstacle were chosen in the past by some ants going in the opposite direction. In other words, when an ant is at the "decision" node, the transitional probabilities would reflect the difference between the alternative trail-to-nest intensities. Similarly, an ant tracing the "food" pheromone would use at the decision node the difference between the alternative trail-to-food intensities created by the returning ants that have traversed either shorter or longer paths around the obstacle in the past. This feature is very important in the beginning — when a new obstacle appears — and the transitional probabilities at the decision node are uniformly distributed. The autocatalytic process is then "kick-started" by the ants going in the opposite direction and using a different pheromone type. The ants going along a shorter return path deposit more pheromone than the ants that select a longer path around an obstacle — simply because

the deposited quantity is inversely proportional to the traversed distance. A higher quantity of pheromone attracts more ants. Eventually, the alternative shortest path is established between a pair of impact nodes.

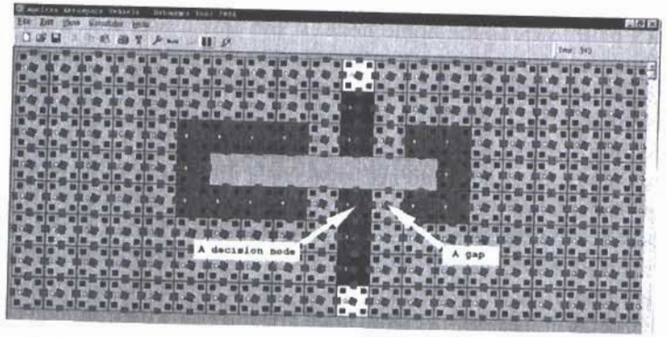
The DRS algorithm uses only one type of pheromone (the "impact" pheromone). Therefore, the ants going in the opposite directions and using the same pheromone type obscure the difference between the alternative trail intensities at the decision node, confounding the choice. For example, a returning ant facing an obstacle ahead and excluding a backtrack possibility has a 50:50 chance of turning left or right, when the trails are not yet established. Choosing a direction at this decision node results in the ant depositing the pheromone either on the left or the right node. Clearly, this deposit is not an informed choice, being driven by a 50:50 chance, and may in fact obscure the pheromone trail. The update of the pheromone on both left and right nodes should, in fact, be done *only* by the ants going in the opposite direction, as these ants have traversed an alternative path. To reiterate, this dilemma is not present when the ants use two types of pheromones. A simple solution enhancing the DRS algorithm, using only one pheromone type, is provided by the "pause" heuristic:

(R4) an ant, facing an obstacle at cycle t and making a transition to the next node, does not deposit any pheromone at cycle t+1, resuming pheromone deposits only from cycle t+2.

The "pause" heuristic initially produces gaps in the trails, next to each decision point (Figure 17). However, these gaps are eventually filled by the ants going in the opposite direction, leading to the reinforcement of the shortest trail. Figures 17-19 illustrate this dynamic with snapshots of the 24 x 8 AAV-CD network array, visualised by the Debugger tool.

The enhanced ADRS algorithm produces rectilinear minimum spanning trees, resulting in reconfigurable impact networks, and performs well in dealing with two well-known problems: the blocking problem and the shortcut problem. *Blocking* occurs when a trail that was found by the ants is no longer available due to an obstacle and an

Figure 17. White cells detected non-critical impacts



An initial vertical trail is destroyed by a horizontal obstacle (seven cells are removed). The returning ants explore two alternative possibilities. The gaps in both trails form next to each decision node.

alternative trail is needed. The *shortcut* corresponds to a new shorter trail becoming available due to repaired cells (Schoonderwoerd, Holland, Brutten, & Rothkrantz, 1997).

Experimental Results

The analysis of algorithm convergence is based on the concept of a connected trail-fragment (CTF). A CTF is a set F of cells with $\varphi \geq \vartheta$ (where ϑ is a given threshold), such that every cell in F is connected with at least one other cell in F, and there exists no cell outside F which is connected to at least one cell in F. We focus here on one important design parameter: pheromone retention rate ψ which determines how much pheromone is left in the cell at the end of each cycle ($\psi = 1.0$ means that there is no evaporation). We carried out 10 experiments with three impacts, for different pheromone retention rates between 0.1 and 0.99. During each experiment, we calculated the average size of CTFs in impact networks, $H(\psi)$, at each time-point, and its standard deviation, $s(y)$, over time. It was observed that low retention rates (for instance, $\psi = 0.86$) lead to chaotic trails; critical retention rates (such as, $\psi = 0.94$) lead to unstable trails ("the edge of chaos"); and high retention rates (for example, $\psi = 0.98$) support stable trails. The critical retention rates between $\psi = 0.90$ and $\psi = 0.94$ result in the most "complex" dynamics: a trail frequently

Figure 18. The gaps of the shorter trail are filled, while the longer trail slowly evaporates without gaps being robustly filled

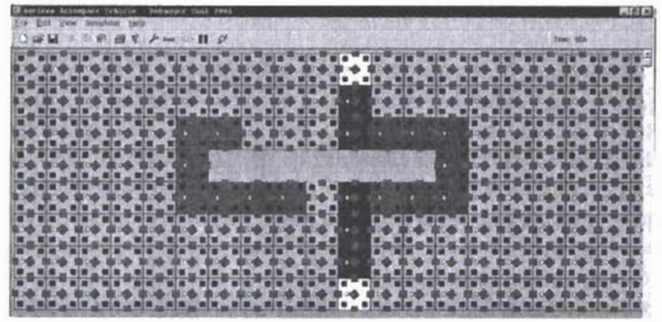
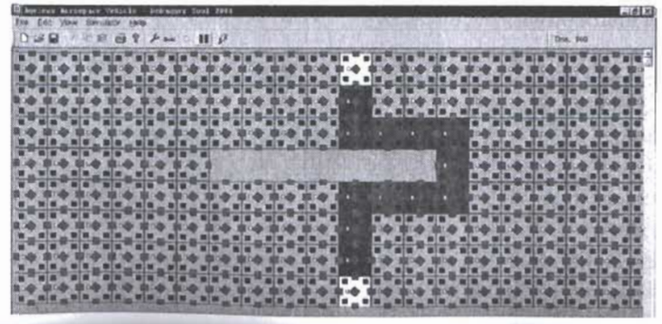


Figure 19. A shorter trail is established



forms and breaks. A detailed picture emerges from Figures 20 and 21. Both plots indicate a clear peak in the standard deviation: $\sigma(\psi)$ peaks in the neighbourhood of $\psi = 0.9$, making the phase transition apparent, and clearly separating ordered and desirable robust scenarios from chaotic and under-performing cases.

Thus, tracing the average size H of CTFs, and its variance σ^2, over time allowed us to identify emergence of rectilinear minimum spanning trees as a phase transition in network connectivity. Our experiments have shown that the ADRS algorithm enhanced with the "pause" heuristic outperforms the original algorithm in terms of these metrics. In particular, we first compared the performance of these two variants in a scenario without obstacles, focusing on the contribution of *adaptive* pheromone reserve quantity and time-to-live counter. The comparison between maximums of $\sigma(\psi)$ for the enhanced ADRS algorithm and the original variant, where the latter was evaluated over three experiments, shows that the ADRS enhancement results in approximately 9% less dispersed data at the edge of chaos (in other words, the standard deviation at its maximum is 2.57 against 2.79, given the same mean size of CTFs), and a more pronounced minimum of $\sigma(\psi)$ after the phase transition, at the retention rate $\psi = 0.96$.

Secondly, we compared the algorithms in a scenario with two impacts and an obstacle, focusing on the contribution of the "pause" heuristics to the solution of the blocking problem. We carried out 10 experiments for each value of the pheromone retention rate in the range between 0.81 and 0.99. During each experiment, a simple straight trail (length 9) was initially formed between two obstacles, and then broken at cycle 200. As before, we calculated the average size of CTFs in impact networks, $H(\psi)$, at each time-point, and its standard deviation, $\sigma(\psi)$, over time. The same three types of dynamics, chaotic, complex, and ordered, were observed. This scenario is more challenging because two "ordered" phases are observed (Figure 21). The first (and the one we are interested in) is the emergence of the stable shorter trail around the obstacle (length 15) as opposed to the longer trail (length 21), followed by the emergence of both stable trails around the obstacle (combined length 29). The first "ordered" phase is separated from the chaotic phase ($\psi < 0.94$) by the "edge of chaos" ($\psi = 0.94$–0.96), and is identified by the minimum of $\sigma(\psi)$, also at the retention rate $\psi = 0.96$. The second "ordered" phase occurs at very high retention rates $\psi \geq 0.99$, and is of no interest: there is enough pheromone to support many trails.

Thus, in terms of solving the blocking problem, the optimal pheromone retention rate ψ can be identified as the one which attains the minimum of the standard deviation $\sigma(\psi)$, following the edge of chaos pointed to by the first maximum of $\sigma(\psi)$, as we increase ψ. When the optimal rate ψ is identified, one can compare the performance of the algorithms at their optima.

The algorithm enhanced with the "pause" heuristics is as good as the main variant in terms of the time it takes for the shorter trail to become the primary choice (on average 147 cycles after the obstacle, for the new algorithm, against 152 cycles for the main variant), and significantly outperformed it in terms of data dispersion both at the edge of chaos and at the optimum:

- the average (over 10 runs) standard deviation at its first edge-of-chaos maximum is 4.92 against 6.08 (24% improvement), and
- the average standard deviation at its first ordered-phase minimum is 3.48 against 5.01 (44% improvement), given the same mean size of CTFs.

Importantly, shorter trails around the obstacle appear as quickly as before but are much more stable with the modified algorithm. In other words, the pause heuristic does not delay emergence of the shorter trail as the primary choice, but makes resultant trails significantly more stable. The main share of the improvements is due to the "pause" heuristics rather than ADRS (which improves dispersion in the order of 10%).

In this section, we considered the emergence of impact network pre-optimising decentralised inspections on an AAV skin, and introduced a new local heuristic improving performance of the modified ACO-DRS algorithm. In summary, the modified algorithm involves one type of non-dissipative evaporating pheromone, simple ant-routing tables containing normalised pheromone values only for immediate neighbours, one type of ant with limited private memory; the dead reckoning scheme, and the transitional probabilities model with obstacle threshold. The ADRS algorithm enhanced with the "pause" heuristic is deployed in the AAV-CD and robustly solves blocking and shortcut problems, producing rectilinear minimum spanning trees for impact-sensing networks.

Figure 20. Background (left z-axis): average size H of CBF's, for different retention rates: $\psi < 0.90$ (chaotic), $\psi = 0.90$–0.94 (unstable), and $\psi > 0.94$ (stable). Foreground (right z-axis, in red colour): standard deviation $\sigma(\psi)$ of the average size $H(\psi)$

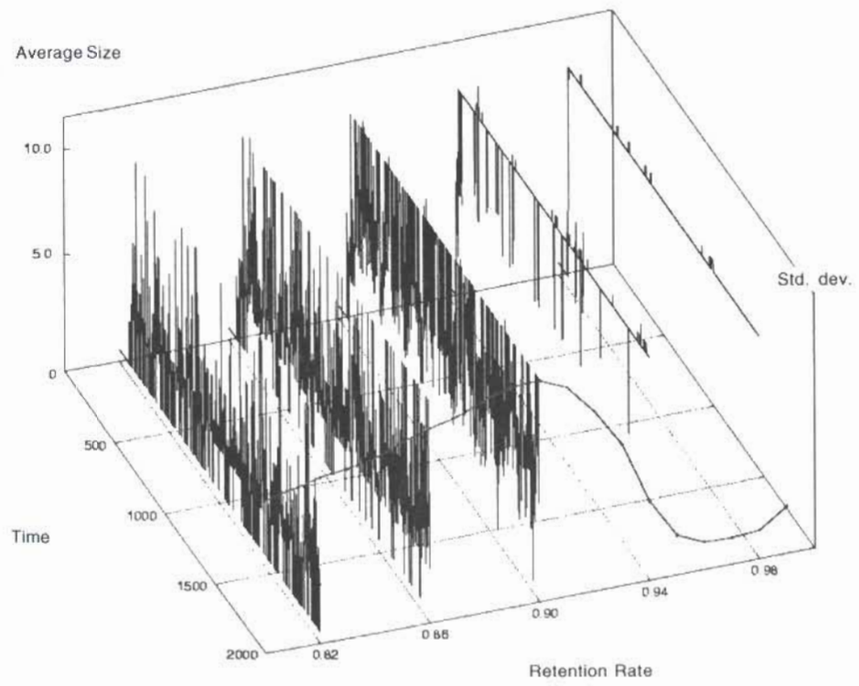


Figure 21. The scenario with three impacts and no obstacles. Standard deviation s of the average size H, for the enhanced ADRS algorithm; phase transition is evident in the range $\psi = 0.90–0.94$

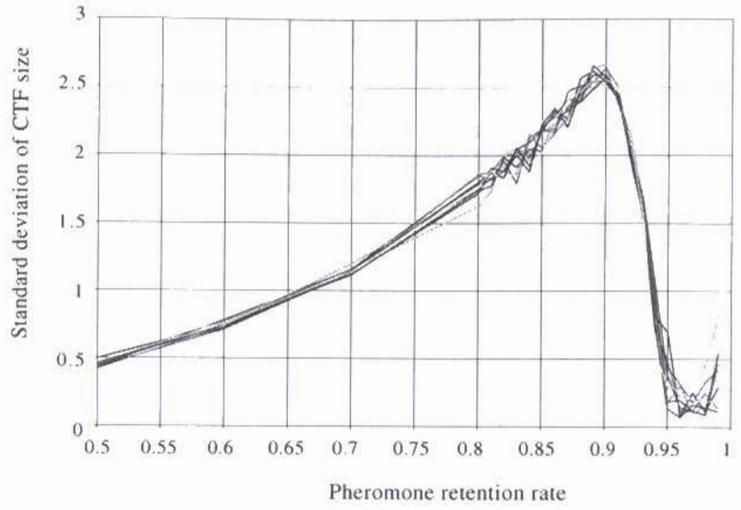
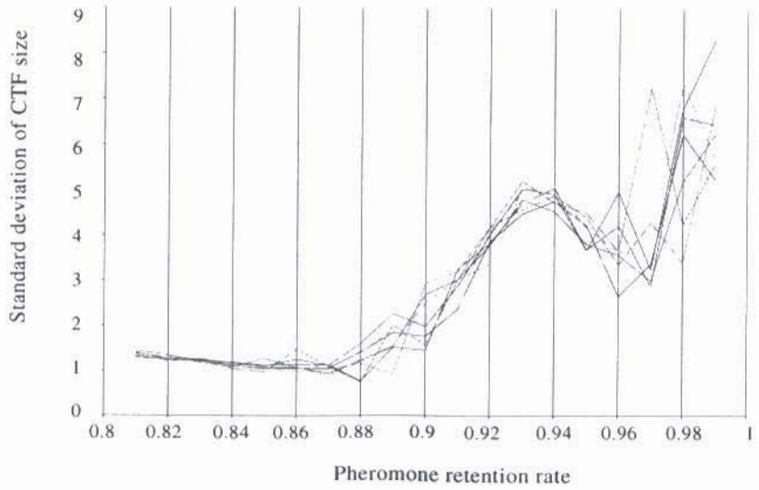


Figure 22. The scenario with two impacts and an obstacle. A chaotic phase (fragmentary trails) is separated by the edge of chaos (first maximum is at $y = 0.94$) from the first ordered phase (stable short trails), followed by another phase transition to combined trails ($\psi > 0.98$)



While we have not evolved parameters for the ACO-DRS algorithm, the observed phase transitions clearly identify the critical values that would be selected by a genetic algorithm (GA) in rewarding stable pheromone trails — similarly to the evolution of impact boundaries guided by the stability metrics. In any case, an optimisation technique should provide a very comprehensive exploration at the edge of chaos. For example, the first phase transition in the dynamics produced by the main variant of our algorithm is in the range $\psi = 0.975–0.977$, and can be easily missed by a GA with inadequate replacement strategies.

DISCUSSION: SELF-ORGANIZATION SELECTION PRESSURES

Self-organising solutions presented in the preceding three sections depend on selection pressures or forces which, through their contribution to the evolutionary fitness functions, constrain the emergent behaviour. One example of a generic selection pressure is the *spatiotemporal stability* of emergent patterns: arguably, any pattern has to be stable before exhibiting another useful task-oriented feature. The sub-critical damage scenario illustrated the use of spatiotemporal stability in evolving impact boundaries (*Impact Boundaries* and *Shape Replication* sections). The impact networks (*Impact Networks and Ant Colonies* section) employ stability as well: the observed phase transitions clearly identify the critical values that would satisfy a fitness function rewarding stable pheromone trails.

Another example of an independent selection force is network *connectivity*, which rewards specific multi-agent network topologies. This force, we believe, is related to both efficiency and robustness, which were identified by Venkatasubramanian, Katare, Patkar, and Mu, (2004) as critical measures underlying optimal network structures. In this context, the efficiency of a graph is defined as the inverse of its average vertex-vertex distance, and is related to the short-term survival. Effective accessibility is defined via a number of vertices reachable from any vertex of a graph component, added over all components. Intuitively, it identifies how quickly a vertex can be reached from other vertices. Structural robustness is then defined with respect to a vertex as the ratio of the effective accessibility of the graph, obtained by deleting this vertex from the original graph, to the maximum possible effective accessibility. Intuitively, this measure captures the importance of this vertex to the connectivity (accessibility) of the graph; in other words, how much the connectivity would be affected if this vertex is removed. Using these definitions, it is possible to define average-case structural robustness as the average computed over all the vertices, or worst-case structural robustness as the minimum computed over all the vertices. Venkatasubramanian et al. (2004) argue that after removal of a vertex, some or all of the sub-graphs could still be functional, and use normalised efficiency of the largest remaining component as an indicator of the functional robustness of the system after damage, relating it to the long-term survival. Both efficiency and robustness identify, in our view, aspects of *connectivity* needed for emergence of optimal multi-agent networks.

Another important selection force is an information-storage ability and *self-referentiality* of representation, providing an emergent pattern with a means for replica-

tion. A well-known example is crystal growth, involving template-based copying process and error correction, and preserving aspects of the crystal structure on macroscopic scales. Each crystal stores a self-referential template (for example, the cross-sectional shape), which may be used in reproduction by splitting. This self-referring arrangement is arguably very simple: the template for growth is the crystal's cross-section, which is directly used in the crystal growth. A much more involved example is the genotype-phenotype relationship, where the degree of self-referentiality is much higher, and the reproduction involves many intermediate steps, mapping genotype into phenotype.

The shape replication process described in the *Shape Replication* section can also be explained in self-referential terms (Prokopenko & Wang, 2004), employing two logical levels. It is well-known that self-replication of a system can be characterised by emergent behaviour and *tangled hierarchies* exhibiting Strange Loops:

> *an interaction between levels in which the top level reaches back down towards the bottom level and influences it, while at the same time being itself determined by the bottom level.* (Hofstadter, 1989)

The shape replication process can be described in these terms as well. An impact boundary emerges at a level which is higher than the object level where individual cells are interacting. The genome of the enclosed multi-cellular shape is a model of an impact boundary, and embedding this higher-level model within every involved cell at the object level is self-referential.

The genome model is obtained by self-inspection of the impact boundary. The process of self-inspection is mirrored by the self-inspection of the genome, carried out internally by each cell at every replication step in order to detect discontinuities in the encoded boundary. Similarly, self-repair of the entire damaged impact-surrounding region is reflected in the internal self-repair of the model (genome). Following Hofstadter's language, the top-level pattern (a boundary) emerges itself out of interactions of cells, while also reaching down to the bottom level and influencing it. This example with self-referential shape replication did not involve explicit metrics for self-referential inspection and repair processes. Nevertheless, our conjecture is that the degree of self-referentiality can be measured and used in evolving multi-agent networks.

Responses to critical damage highlight the role of a low computation and communication complexity as another selection force. The main principle in considering emergency and/or "panic" responses is that the system needs to alter its priorities from long-term survival to emergency short-term survival, on many levels. In terms of the AAV, an emergency response may therefore require changing priorities of communication messages, an increase in the rate of polling the buffers of the communication ports, redirection of more power to specific modules, while temporarily disabling other modules, and so on. Subsequently, it may cause an activation of secondary passive and mobile sensors. This cascading scenario requires a fast and unconscious (un-reasoned) reaction, immediately upon a detection of a specific sensory input (trigger). This trigger should be detected locally, simply because detecting and matching near-simultaneous remote sensory inputs would have to be done "deeper" within the system, leaving less time for the emergency response. In other words, the trigger is "locally-situated" both in space and time, and the selection pressure rewarding a low computation and communication complexity would guide an evolution of adequate responses.

CONCLUSION

This chapter has presented an approach to the structural health management (SHM) of future aerospace vehicles that will need to operate robustly in very adverse environments. Such systems will need to be intelligent and to be capable of self-monitoring and ultimately, self-repair. The robustness requirement is best satisfied by using a distributed rather than a centralised system, and this has been assumed from the outset. Networks of embedded sensors, active elements, and intelligence have been selected to form a prototypical "smart skin" for the aerospace structure, and a methodology based on multi-agent networks developed for the system to implement aspects of SHM by processes of self-organisation. This has been developed in the context of a hardware test-bed, the CSIRO/NASA "concept demonstrator" (CD), a cylindrical structure with a metallic smart skin with 196 sensor/actuator/processor modules. A number of SHM algorithms related to damage detection and assessment have been developed and tested on this demonstrator.

A future aerospace vehicle will be expected to respond to a variety of damage situations which, moreover, vary with time and circumstance. Designing a general system with distributed intelligence which can self-organise solutions to many different problems is a very difficult task which we have simplified considerably by dividing the problems into manageable components as described in the *Response Matrix Approach* section, then seeking self-organising solutions to each component. This top-down/bottom-up (TDBU) approach allows solutions to be achieved whilst retaining the flexibility and emergent behaviour expected from complex multi-agent networks.

This breakdown of problems into components was achieved with the aid of a "response matrix" (Table 1) and three significant scenarios were analysed in this fashion. These were (a) critical damage, which threatens the integrity of the vehicle, (b) sub-critical damage, which requires immediate action although is not life-threatening, and (c) minor damage, whose cumulative effects need to be monitored and acted on when appropriate. From these scenarios, three main components were selected, and self-organising solutions developed for each and tested on the hardware test-bed. These components were: (1) the formation of "impact boundaries" around damage sites, allowing the extent of any damage to be assessed and communicated to other parts of the vehicle; (2) self-assembling "impact networks", robust communications links which connect damage sites, enabling inspection of minor damage; and (3) shape replication, a demonstration of an autonomous repair mechanism by which the network "grows", at a remote site, a new region of the correct shape to replace a damaged area. The first two of these have been successfully implemented on the hardware test-bed, giving confidence in the feasibility of the overall approach.

Future work will continue with the implementation of other necessary components, such as detailed diagnosis and prognosis for the different damage scenarios (Prokopenko et al., 2005b), sensor-data clustering (Mahendra, Prokopenko, Wang, & Price, 2005; Prokopenko, Mahendra, & Wang, 2005), robust communication to action-initiating sites (Li, Guo, & Poulton, 2004) and actions aimed at repair or mitigation of damage. One such development, currently in the preliminary stages, aims at developing a means of secondary inspection, an independent system which, when invoked by a report of possible damage, is capable of examining the relevant site and assessing the extent of the damage.

Looking further ahead, it is clear that the functionalities of sensing, computation, and action must merge with the material properties of the vehicle, moving closer to the real meaning of a smart skin. Although it may be some time before such a development is fully realised, recent progress in materials science and nano-technology gives confidence that it is achievable. We believe that the basic approach outlined in this chapter, of seeking self-organising solutions to critical components within an intelligent multi-agent framework, will still form the backbone of such future developments.

ACKNOWLEDGMENTS

A part of this work was carried out under CSIRO contract to NASA. It is a pleasure to record our appreciation to Dr. Ed Generazio (NASA Langley Research Center) for his encouragement and support. The authors are grateful to other members of the AAV project for their contributions to this work.

REFERENCES

Abbott, D., Ables, J., Batten, A., Carpenter, D. C., Collings, A. F., Doyle, B., et al. (2003, January). *Development and evaluation of sensor concepts for ageless aerospace vehicles, Rep. 3: Design of the concept demonstrator* (Conf. Rep. No. TIPP 1628). CSIRO Telecommunications and Industrial Physics.

Abbott, D., Doyle, B., Dunlop, J., Farmer, T., Hedley, M., Herrmann, J., et al. (2002). *Development and evaluation of sensor concepts for ageless aerospace vehicles: Development of concepts for an intelligent sensing system* (NASA Tech. Rep. No. NASA/CR-2002-211773). Hampton, VA: Langley Research Center.

Batten, A., Dunlop, J., Edwards, G., Farmer, T., Gaffney, B., Hedley, M., et al. (2004, April). *Development and evaluation of sensor concepts for ageless aerospace vehicles, Rep. 5, Phase 2: Implementation of the concept demonstrator* (Conf. Rep. No. TIPP 2056). CSIRO Telecommunications and Industrial Physics.

Buhl, J., Deneubourg, J. -L., & Theraulaz, G. (2002, September 12-14). Self-organized networks of galleries in the ant messor sancta. In M. Dorigo, G. Di Caro, & M. Sampels (Eds.), *Proceedings of the 3rd International Workshop on Ant Algorithms — ANTS2002*, Brussels, Belgium (pp. 163-175).

Camazine, S., Deneubourg, J. -L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self-organization in biological systems*. Princeton, NJ: Princeton University Press.

Colorni, A., Dorigo, M., & Maniezzo, V. (1992). Distributed optimization by ant colonies. In F. S. Varela & P. Bourgine (Eds.), *Toward a Practice of Autonomous Systems: Proceedings of the 1st European Conference on Artificial Life*, Paris, December 11-13, 1991 (pp. 134-142).

Crutchfield, J. (1994). The calculi of emergence: Computation, dynamics, and induction. *Physica D, 75*, 11-54.

Deneubourg, J. -L., & Goss, S. (1989). Collective patterns and decision making. *Ethology, Ecology & Evolution, 1*, 295-311.

Dorigo, M., & Di Caro, G. (1999, July 6-9). Ant colony optimization: A new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC (pp. 1470-1477). Piscataway, NJ: IEEE Press.

Dorigo, M., Maniezzo, M., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 26*(1), 1-13.

Durbeck, L. J. K., & Macias, N. J. (2002, July 29-August 1). Defect-tolerant, fine-grained parallel testing of a cell matrix. *In Proceedings of SPIE, Series 4867 — Convergence of IT and Communications (ITCom)*, Boston (pp. 71-85).

Eppstein, D. (1996). *Spanning trees and spanners* (Tech. Rep. No. 96-16). Department of Information and Computer Science, University of California, Irvine. Also in J.-R. Sack & J. Urrutia (Eds.), *Handbook of computational geometry* (pp. 425-461). Amsterdam: Elsevier.

Ferber, J. (1999). *Multi-agent systems*. Reading, MA: Addison Wesley.

Foreman, M., Prokopenko, M., & Wang, P. (2003, Septebmer 14-17). Phase transitions in self-organising sensor networks. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, & J. Ziegler (Eds.), *Advances in Artificial Life — Proceedings of the 7th European Conference on Artificial Life (ECAL-03)*, Dortmund, Germany (pp. 781-791). Heidelberg: Springer.

Generazio, E. (1996). Nondestructive evaluation science in the year 2015+. *NASA Langley Research Center, Nondestructive Evaluation Sciences Branch*. Retrieved from http://nesb.larc.nasa.gov/nde2015.html

Gulati, S., & Mackey, R. (2003, July). *Prognostics methodology for complex systems: Automatic method to detect and react to complex degradation and incipient faults* (Tech. Rep. No. NPO-20831). NASA Jet Propulsion Laboratory, Pasadena, California.

Hedley, M., Hoschke, N., Johnson, M., Lewis, C., Murdoch, A., Price, D., et al. (2004, December 5-8). Sensor network for structural health Monitoring. *In Proceedings of the International Conference on Intelligent sensors, Sensor Networks and Information Processing (ISSNIP-2004)*, Melbourne, Australia (pp. 361-366).

Hofstadter, D. R. (1989). *Gödel, Escher, Bach: An eternal golden braid*. New York: Vintage Books.

Holland, O., & Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life, 5*, 173-202.

Kahng, A. B., & Robins, G. (1995). *On optimal interconnections for VLSI*. Boston: Kluwer Academic Publishers.

Klyubin, A. S., Polani, D., & Nehaniv, C. L. (2004, June 24-26). Organization of the information flow in the perception-action loop of evolved agents. *Proceedings of the NASA/DoD Conference on Evolvable Hardware*, Seattle, WA (pp. 177-180). Piscataway, NJ: IEEE Computer Society Press.

Li, J., Guo, Y., & Poulton, G. (2004, December 4-6). Critical damage reporting in intelligent sensor networks. *Proceedings of the 17th Joint Conference on Artificial Intelligence (AI'04)*, Cairns, Australia (pp. 26-38).

Lovatt, H., Poulton, G., Price, D., Prokopenko, M., Valencia, P., & Wang, P. (2003, July 14-18). Self-organising impact boundaries in ageless aerospace vehicles. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, & M. Yokoo (Eds.), *Proceedings of the*

2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia (pp. 249-256). New York: ACM Press.

Mahendra Rajah, P., Prokopenko, M., Wang, P., & Price, D. (2005, September 14-16). Towards adaptive clustering in self-monitoring multi-agent networks. *Proceedings of the 9th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES-2005)*, Melbourne, Australia.

Mange, D., Sanchez, E., Stauffer, A., Tempesti, G., Marchal, P., & Piguet, C. (1998). Embryonics: A new methodology for designing field-programmable gate arrays with self-repair and self-replicating properties. *IEEE Transactions on VLSI Systems, 6*, 387-399.

Price, D., Scott, A., Edwards, G., Batten, A., Farmer, A., Hedley, M., et al. (2003, September). An integrated health monitoring system for an ageless aerospace vehicle. In F.-K. Chang (Ed.), *Structural Health Monitoring 2003 — Proceedings of 4th International Workshop on Structural Health Monitoring*, Stanford, CA (pp. 310-318). Lancaster, PA: D. E. Stech Publications.

Prokopenko, M., Mahendra, P., & Wang, P. (2005, September 5-9). On convergence of dynamic cluster formation in multi-agent networks. *Proceedings of the 8th European Conference on Artificial Life (ECAL 2005)*, Canterbury, Kent, UK.

Prokopenko, M., & Wang, P. (2004, September 12-15). On self-referential shape replication in robust aerospace vehicles. *Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, Boston (pp. 27-32).

Prokopenko, M., Wang, P., & Price, D. (2005, June 29-July 1). Complexity metrics for self-monitoring impact sensing networks. In *Proceedings of the NASA/DoD International Conference on Evolvable Hardware (EH-05)*, Washington, DC.

Prokopenko, M., Wang, P., Price, D., Valencia, P., Foreman, M., Farmer, A. (2005a). Self-organising hierarchies in sensor and communication networks. *Artificial Life (special issue on Dynamic Hierarchies), 11*(4).

Prokopenko, M., Wang, P., Scott, D. A., Gerasimov, V., Hoschke, N., & Price, D. C. (2005b, Septebmer 14-16). On self-organising diagnostics in impact sensing networks. *Proceedings of the 9th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES-2005)*, Melbourne, Australia.

Prosser, W. H., Allison, S. G., Woodard, S. E., Wincheski, R. A., Cooper, E. G., Price, D. C., et al. (2004, December 16-17). Structural health management for future aerospace vehicles. *Proceedings of the 2nd Australasian Workshop on Structural Health Monitoring (2AWSHM)*, Melbourne, Australia (pp. 1-15).

Schoonderwoerd, R., Holland, O. E., Brutten, J., & Rothkrantz, L. J. M. (1997). Ant-based load balancing in telecommunication networks. *Adaptive Behaviour, 5*(2), 169-207.

Schweitzer, F., & Tilch, B. (2002). Self-assembling of networks in an agent-based model. *Physical Review E, 66*, 1-9.

Sipper, M., Mange, D., & Stauffer, A. (1997). Ontogenetic hardware. *BioSystems, 44*, 193-207.

Trianni, V., Labella, T. H., Gross, R., Sahin, E., Dorigo, M., & Deneubourg, J. -L. (2002). *Modelling pattern formation in a swarm of self-assembling robots* (Tech. Rep. No. TR/IRIDIA/2002-12, May). Bruxelles, Belgium: Institut de Recherches Interdisciplinaires et de Developpement en Intelligence Artificielle.

Venkatasubramanian, V., Katare, S., Patkar, P. R., & Mu, F. -P. (2004). Spontaneous emergence of complex optimal networks through evolutionary adaptation. *Computers and Chemical Engineering, 28*(9), 1789-1798.

Wang, P., & Prokopenko, M. (2004, July 13-17). Evolvable recovery membranes in self-monitoring aerospace vehicles. In *Proceedings of the 8th International Conference on Simulation of Adaptive Behaviour (SAB-2004)*, Los Angeles, CA (pp. 509-518).

Wang, P., Valencia, P., Prokopenko, M., Price, D., & Poulton, G. (2003, June 30-July 3). Self-reconfigurable sensor networks in ageless aerospace vehicles. In U. Nunes, A. T. de Almeida, A. K. Bejczy, K. Kosuge, & J. A. T. Machado (Eds.), *Proceedings of the 11th International Conference on Advanced Robotics (ICAR-03)*, Coimela, Portugal (pp. 1098-1103).

Wessnitzer, J., Adamatzky, A., & Melhuish, C. (2001, September 10-14). Towards self-organising structure formations: A decentralized approach. In J. Kelemen & P. Sosík (Eds.), *Advances in Artificial Life: Proceedings of the 6th European Conference on Artificial Life*, Prague, Czech Republic (pp. 573-581).

Wu, Y. -F., Widmayer, P., Schlag, M. D. F., & Wong, C. K. (1987). Rectilinear shortest paths and minimum spanning trees in the presence of rectilinear obstacles. *IEEE Transactions on Computers, C-36*, 321-331.